2.1 INDEPENDENT COMPONENT ANALYSIS

2.1.1 Introduction

One of the statistical models to solve BSS problem is using Independent Component Analysis (ICA). (Hyvarinen et al 2001). ICA is a technique that recovers independent sources for a given sensor observations that are unknown linear mixtures of the unobserved independent source signals. ICA is a way of finding a linear non-orthogonal co-ordinate system in any multivariate data. The directions of the axes of this co-ordinate system are determined by both the second and higher order statistics of the original data. The goal of ICA is to perform a linear transform which makes the resulting variables as statistically independent from each other as possible. ICA is a generally applicable method to several challenges in signal processing. It reveals a diversity of theoretical questions and opens a variety of potential applications. Two different research communities have considered the analysis of independent components. One group has analyzed the classical and difficult signal processing problem, which is the separation of mixed sources observed in an array of sensors. In parallel to this analysis, unsupervised learning rules based on information theory are proposed by another group of researchers.
ICA is a mathematical model for separating a signal into its most probable additive components assuming the statistical independence. In ICA, prior information on the statistical properties of the source signals $S_i(t)$ are used to estimate the mixing weight matrix. It is assumed that the two source signals $S_1(t)$ and $S_2(t)$ at each time instant ‘t’ are statistically independent of each other. ICA is originally developed to deal with blind source separation problems that are closely related to cocktail-party problem and it is a very general purpose method of signal processing and data analysis. Another application of ICA is feature extraction. The ICA is able to solve the BSS problem under the following assumptions (Comon 1994).

(i) The sources $M_i$ are statistically independent.
(ii) The sources must have non-Gaussian distributions.
(iii) Each measured signal is a linear combination of each of the independent signals.
(iv) There are an equal number of measured and independent signals.

This existing model has been first reviewed and then the simulation of this model has been carried out. Each of the original independent signals is denoted as ‘$S_i$’ and linearly combined measured (or observed) signals as ‘$O_i$’. ‘$O$’ is a column vector of ‘n’ observed signals. Each measured signal is expressed as a linear combination of the original independent signals, given by the equation

$$O_i = a_1S_1 + a_2S_2 + ... + a_nS_n$$  \hspace{1cm} (2.1)

The entire system of ‘n’ measured signals is expressed as

$$O = A.S$$  \hspace{1cm} (2.2)
where each row of ‘O’ is a set of readings of each signal \(O_i\); each row of ‘S’ is an original signal \(S_i\) and ‘A’ is an \(n \times n\) mixing matrix that generates ‘O’ from ‘S’. The goal of ICA is to find ‘S’ and ‘A’ for a given ‘O’. At first glance, this problem seems severely under constrained. However, ICA is looking for specific features in ‘S’ that allows the signals to be separated. Prior information on the statistical properties of the source signals \(S_i(t)\) are used to estimate the mixing weight matrix. It is enough to assume that the two source signals \(S_1(t)\) and \(S_2(t)\) at each time instant ‘t’ are statistically independent. The ICA is applied wherever there are ensembles of multivariate data, e.g., where Principal Components Analysis (PCA) technique has been used. Examples include blind separation of mixed speech signals, biomedical data processing of EEG data (Makeig et al 1996) and finding features in data (e.g., learning edge detectors for ensembles of natural images).

2.1.2 Defining Independence

Random variables ‘X’ and ‘Y’ are said to be independent if the conditional probability of ‘X’ with respect to ‘Y’ is just the probability of ‘X’. In other words, knowing a value of ‘Y’ gives nothing about ‘X’. This is expressed as

\[
P(X/Y) = P(X) \tag{2.3}
\]

Since \(P(X/Y) = P(X, Y)/P(Y)\), where \(P(X, Y)\) is the joint density function of ‘X’ and ‘Y’ and it is given by

\[
P(X, Y) = P(X) \times P(Y) \tag{2.4}
\]

Another important feature of statistical independence is

\[
\text{mean}(g_1(X) \times g_2(Y)) = \text{mean}(g_1(X)) \times \text{mean}(g_2(Y)) \tag{2.5}
\]
for any functions ‘g₁’ and ‘g₂’ and X ≠ Y.

The covariance between ‘X’ and ‘Y’ is expressed as

$$\text{cov}(X, Y) = \text{mean}(X \times Y) - (\text{mean}(X) \times \text{mean}(Y))$$  \hspace{1cm} (2.6)

For a random variable ‘X’, cov(X,X) is equal to the variance of ‘X’. If ‘X’ and ‘Y’ are independent, then mean(X×Y) = mean(X)×mean(Y) and the covariance is zero. In source separation, prior knowledge is statistical independence of ‘S’. The covariance of two statistically independent variables is always zero. The converse is not always true. Just because the covariance is zero, it does not mean that the sources ‘A’ and ‘B’ are independent. However, in the special case of Gaussian variables, zero covariance does imply independence. This feature of Gaussian variables is used to find columns of ‘W’ in W.X=S. Previously, it has been stated that each measured signal in ‘O’ is a linear combination of the independent signals in ‘S’. The mixing matrix ‘A’ is invertible such that A⁻¹=W and each of the independent components in ‘S’ is also expressed as a linear combination of the measured signals in ‘O’ (S=W.O).

2.1.3 Drawbacks of ICA

- ICA is restricted to linear mixtures of sources signals, which is nevertheless a reasonable assumption for many applications. Such a mixture is described by the equation \( x = Hs + v \).

- ICA requires more than one simultaneously recorded mixture in order to find the individual signals in any one mixture. It is worth stressing here that ICA does not incorporate any knowledge
specific to speech signals; in order to work, it requires simply that the individual voice signals are unrelated. A critical caveat is that they require at least as many mixtures as there are source signals.

- It can not recover the real scale of source signals.

- One of the assumptions of ICA is that the source signals must be uncorrelated with each other. The success of ICA in a given application depends on the validity of the assumptions on which ICA is based. These assumptions are violated to some extent by most data sets. ICA method fails when its assumptions (i.e. linear mixing and independence) are severely violated.

- The asymptotic behavior of an ICA algorithm should depend on the ‘true’ probability density function of each source, but in practical applications these densities are most often unknown.

Despite its great success in many real world applications, the classical ICA model has several limitations such as mentioned above, which have motivated new learning paradigms that are beyond the classical setting. So, there are many real world applications that require new algorithms for source separation.

### 2.1.4 Simulation Result

The three audio signals which are given in Table 2.1 are mixed by the mixing matrix ‘A_ica’, generated randomly by the Matlab function \texttt{rand}(). Maximum iteration is set at 5000 and the learning rate parameter, $\eta = 0.1$. The maximum mask length is set at 500. The mixture signal is given as
input to ICA code. The original, mixture and separated output signals are shown in Figure 2.1. The random mixing matrix $A_{ica}$ is given by

$$A_{ica} = \begin{bmatrix} -1.1207 & -1.1195 & -0.0629 \\ -0.8123 & 0.6140 & 0.7946 \\ 0.7862 & 1.8642 & -0.2446 \end{bmatrix}$$ (2.7)

To recover source signals, the variables short and long half lives used in the ICA code are set at 1 and 900000 respectively. Initial weight matrix $W_0$ is given by

$$W_0 = \begin{bmatrix} -0.4893 & -0.2606 & 0.8322 \end{bmatrix}$$ (2.8)

**Table 2.1 Example of source signals used for simulation**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Source 1 (beans.wav)</th>
<th>Source 2 (wsong2.wav)</th>
<th>Source 3 (ssong1.wav)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit rate</td>
<td>64 Kbps</td>
<td>48 Kbps</td>
<td>176 Kbps</td>
</tr>
<tr>
<td>Sample size</td>
<td>8-bit mono</td>
<td>8-bit mono</td>
<td>8-bit mono</td>
</tr>
<tr>
<td>Sample rate</td>
<td>8 KHz</td>
<td>6 KHz</td>
<td>22 KHz</td>
</tr>
<tr>
<td>No. of samples</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

Using gradient ascent algorithm, the correlation of a single source with sources extracted by the initial weight matrix $W_0$ is obtained as

$$corr_{\text{sin}} = \begin{bmatrix} 0.9997 & 0.0265 & 0.0236 \end{bmatrix}$$ (2.9)
Using gradient ascent algorithm, correlations between sources and all recovered signals are obtained as

\[
\text{corr}_{\text{all}} = \begin{bmatrix}
0.9997 & 0.0265 & 0.0236 \\
0.0087 & 0.9987 & 0.0584 \\
0.0179 & 0.0391 & 0.9992
\end{bmatrix}
\]  

(2.10)

It has been observed from the Figure 2.1 that the mean value of the recovered signals by ICA algorithm is found to be 0.035, whereas the mean value of the original signal is 0.72. In this experiment, it is observed that the amplitudes of the recovered source signals by ICA algorithm are smaller by 0.685 (mean value) than that of the original source signals. Therefore, it is concluded that the ICA algorithm has scaling problem, which has been eliminated by the proposed USGDA algorithm (as explained in sections 4.2.1, 4.2.2, 4.2.3 and 4.2.4)

2.2 MEASURE OF SEPARATION QUALITY OF RECOVERED SIGNALS BY ICA

2.2.1 Performance Index (PI)

After the separating matrix ‘W’ has been computed by the ICA algorithm, the separation quality (Amari et al 1996) is measured by means of performance index (PI), given by

\[
\text{PI} = \frac{1}{m} \sum_{i=1}^{m} \left\{ \sum_{j=1}^{m} \frac{|s_{ij}|^2}{\max_q |s_{qj}|} - 1 \right\} + \frac{1}{m} \sum_{j=1}^{m} \left\{ \sum_{i=1}^{m} \frac{|s_{ij}|^2}{\max_q |s_{iq}|} - 1 \right\}
\]

(2.11)

where \(s_{ij}\) denotes the \(i^{th}\) row and \(j^{th}\) column of \(P = A.W\). Lower performance index indicates good separation quality. Ideally, it should be zero.
Figure 2.1 Waveforms of ICA algorithm
It has been observed from Table 2.2 and Figure 2.2 that the performance index of ICA algorithm is decreased from 18.0 to 0.5 for iterations increased from 100 to 823.

Figure 2.2 Performance Index of ICA algorithm over 823 independent runs

### 2.2.2 Signal to Noise Ratio

Another measure of separation quality (Bedoya et al 2003) used is the Signal to Noise Ratio (SNR) of the separated outputs, given by

$$SNR = 10\log_{10} \left( \frac{\sum m(t)^2}{\sum n(t)^2} \right)$$  \hspace{1cm} (2.12)
where \( m(t) \) is the desired signal and \( n(t) = o(t) - m(t) \) is the noise indicating the undesired signal. \( o(t) \) is the estimated source signals. It has been observed from Table 2.2 and Figure 2.3 that SNR of the separated signals by the ICA algorithm is varied from 5.7 to 19.0 for sampled data increased from 100 to 1000.

Table 2.2  Performance index and Signal to noise ratio of recovered signals using ICA algorithm

<table>
<thead>
<tr>
<th>No. of Iterations</th>
<th>Performance Index (PI)</th>
<th>Signal to Noise Ratio (SNR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>18.0</td>
<td>5.7</td>
</tr>
<tr>
<td>200</td>
<td>16.0</td>
<td>7.2</td>
</tr>
<tr>
<td>300</td>
<td>14.0</td>
<td>9.0</td>
</tr>
<tr>
<td>400</td>
<td>11.5</td>
<td>11.4</td>
</tr>
<tr>
<td>500</td>
<td>8.3</td>
<td>12.6</td>
</tr>
<tr>
<td>600</td>
<td>6.9</td>
<td>13.1</td>
</tr>
<tr>
<td>700</td>
<td>3.5</td>
<td>14.0</td>
</tr>
<tr>
<td>800</td>
<td>1.6</td>
<td>16.0</td>
</tr>
<tr>
<td>823</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>900</td>
<td>-</td>
<td>17.8</td>
</tr>
<tr>
<td>1000</td>
<td>-</td>
<td>19.0</td>
</tr>
</tbody>
</table>
2.3  BSS USING TEMPORAL PREDICTABILITY (JTP)

2.3.1  Introduction

BSS using temporal predictability (JTP) is another conventional method to solve BSS problem. In this method, the covariance of sources is first determined. The covariance of two or more statistically independent variables is always zero. The converse is not always true i.e., just because the covariance is zero, it does not mean that the two sources ‘A’ and ‘B’ are independent. However, in the special case of Gaussian variables, zero covariance does imply independence. This feature of Gaussian variables is used to find columns of ‘W’ in $W.X = S$ (Stone 2001).

BSS using temporal predictability algorithm separates the original source signals by simultaneous diagonalization of long and short term mixture covariance matrices. In this method, the following assumptions are made.

(i)  The mixing matrix is non-singular.
(ii) The sources are spatially uncorrelated and second-order non-stationary.

With these assumptions, it is shown that the simultaneous diagonalization of the long and short term mixture covariance matrices allows to estimate the demixing matrix ‘W’. The following algorithm is used to recover source signals.

Step 1. Set short and long half lives.
Step 2. Set short term mask and long term mask to filter out each column in the mixing matrix.
Step 3. Filter each column of mixtures array.
\[ S = \text{filter}(s\_\text{mask}, 1, \text{mixtures}); \]
\[ L = \text{filter}(l\_\text{mask}, 1, \text{mixtures}); \]
Step 4. Find covariance matrices.
\[ U = \text{cov}(S, 1); \]
\[ V = \text{cov}(L, 1); \]
Step 5. Find Eigenvectors W and Eigenvalues d.
\[ [W, d] = \text{eig}(V, U); \]
Step 6. Recover source signals.
Step 7. Plot results.
Step 8. Rescale the output vector to zero means and unit variance for obtaining the recovered signals.

This algorithm has been reviewed and its simulation result is shown in Figure 2.4. Its performance has been compared with proposed USGDA algorithm with respect to the performance index and the signal to noise ratio.
2.3.2 Drawbacks of JTP

- This method can also separate only linearly mixed signals and also requires more than one simultaneously recorded mixture in order to find the individual signals in any one mixture.

- Like, ICA method, it also can not recover the real scale of source signals.

- The method is based on the assumption that different source signals are associated with distinct critical points in signal predictability ‘F’. However, if any two source signals have the same degree of predictability ‘F’, then two eigenvectors $W_i$ and $W_j$ have equal eigenvalues (and are associated with the same critical points in F). Therefore, any vector $W_k$ that lies in the plane defined by $W_i$ and $W_j$ also maximizes ‘F’, but $W_k$ can not be used to extract a source signal.

- In this method, a measure of temporal predictability alone is used to separate the original source signals. No formal proof is given for the temporal predictability conjecture.

2.3.2 Simulation Result

Each audio signal given in Table 2.1 is set to have zero mean and unit variance by the Matlab functions `mean()` and `std()`. They are mixed by the mixing matrix ‘a_jtp’ given in Equation (2.13) and each column of mixtures array are filtered by the Matlab function `filter()` to recover source signals and the covariance matrices are then found by the Matlab function `cov()`. To determine the inverse of mixing matrix, eigenvalues of
the covariance matrices are found by the `Matlab` function `eig( )`. Finally, the sources are recovered by multiplying the mixture signal with the inverse matrix

\[
\mathbf{a}_{\text{jtp}} = \begin{bmatrix}
-0.3445 & -0.8557 & 0.6517 \\
-0.4771 & 0.3918 & 1.2966 \\
1.3739 & -1.0283 & 1.1385
\end{bmatrix}
\] (2.13)

In JTP algorithm, the correlations between sources and all the recovered signals are obtained as

\[
\mathbf{corr}_j = \begin{bmatrix}
0.0414 & 0.1759 & 0.9835 \\
0.9967 & 0.0605 & 0.0537 \\
0.0457 & 0.9625 & 0.2673
\end{bmatrix}
\] (2.14)

It has been observed from the Figure 2.4 that the mean value of the recovered signals by JTP algorithm is found to be -1.063, whereas the mean value of the original signal is 0.412. In this experiment, it is observed that the amplitudes of the recovered source signals by JTP algorithm are smaller by 0.651 (mean value) than that of the original source signals. Therefore, it is concluded that the ICA algorithm has a scaling problem, which has been eliminated by the proposed USGDA algorithm (as explained in sections 4.2.1, 4.2.2, 4.2.3 and 4.2.4).
Figure 2.4 Waveforms of JTP algorithm
2.4 MEASURE OF SEPARATION QUALITY OF RECOVERED SIGNALS BY JTP

2.4.1 Performance Index

After the separating matrix ‘W’ has been computed by the BSS using temporal predictability (JTP) algorithm, the separation quality is measured by the performance index determined by the Equation (2.11). Lower performance index indicates good separation quality. Ideally, it should be zero. It has been observed from Table 2.3 and Figure 2.5 that the performance index of JTP algorithm is decreased from 16.0 to 0.3 for iterations increased from 100 to 823.

Figure 2.5 Performance index of JTP algorithm over 823 independent runs
2.4.2 Signal to Noise Ratio

Another measure of separation quality used is the Signal to Noise Ratio (SNR) of the separated outputs determined by the Equation (2.12). It has been observed from Table 2.3 and Figure 2.6 that SNR of the separated signals by the JTP algorithm is varied from 5.2 to 16.6 for sampled data increased from 100 to 1000.

So, it is concluded that both statistical methods suffer from scaling problem and the signal to noise ratio and performance index of these two algorithms are to be improved and they can separate only linearly mixed signals whereas the Artificial Neural Networks can be used to solve BSS problem in which the signals are nonlinearly mixed since they exhibit nonlinear input/output mapping capabilities and can be trained with examples of a problem by unsupervised algorithms to enable the network to acquire knowledge about it.

ANNs can predict new outcomes from past trends and they can learn by examples. Moreover, they are fault tolerant and robust and they process information in parallel, at high speed and in a distributed manner. Therefore, an attempt is made to separate the source signals by BPN network. The major limitation of the backpropagation network is its slow convergence time and it may end up with local minima. Moreover, there is no proof of convergence, although it seems to perform well in practice. Due to stochastic gradient descent on a nonlinear error surface, it is likely that most of the time the result may converge to some local minimum on the error surface. Another major limitation is the problem of scaling and when the complexity of the problem is increased, there is no guarantee that good generalization would result. So, the network size, such as the number of hidden layers has to be increased.
Table 2.3  Performance index and Signal to noise ratio of separated signals using JTP algorithm

<table>
<thead>
<tr>
<th>No. of Samples</th>
<th>Performance Index (PI)</th>
<th>Signal to Noise Ratio (SNR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>16.0</td>
<td>5.2</td>
</tr>
<tr>
<td>200</td>
<td>14.0</td>
<td>7.1</td>
</tr>
<tr>
<td>300</td>
<td>12.0</td>
<td>8.5</td>
</tr>
<tr>
<td>400</td>
<td>10.2</td>
<td>10.6</td>
</tr>
<tr>
<td>500</td>
<td>7.6</td>
<td>11.2</td>
</tr>
<tr>
<td>600</td>
<td>6.4</td>
<td>12.0</td>
</tr>
<tr>
<td>700</td>
<td>2.8</td>
<td>12.5</td>
</tr>
<tr>
<td>800</td>
<td>1.0</td>
<td>14.9</td>
</tr>
<tr>
<td>823</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>900</td>
<td>-</td>
<td>15.3</td>
</tr>
<tr>
<td>1000</td>
<td>-</td>
<td>16.6</td>
</tr>
</tbody>
</table>

Figure 2.6  Signal to noise ratio of JTP algorithm
This results in heavy burden on the computation and network complexity. Moreover, BPN uses a black box fitting approach that causes the MLP to suffer from the ‘overfitting’ phenomenon and the global mapping problem in which the hidden neurons tend to interact globally with each other. This results in an improvement at one point, but deterioration at some other points. (Tan et al 2001).

2.5 MOTIVATION FOR USING RBF NEURAL NETWORKS

The BPN and RBF neural networks differ from each other in several important respects given below, which motivate to use RBF for BSS problem.

i. A RBF network has a single hidden layer, whereas a BPN may have one or more hidden layers.

ii. Typically, the computation nodes of a BPN, located in a hidden or an output layer, share a common neuronal model. On the other hand, the computation nodes in the hidden layer of an RBF network are quite different and serve a different purpose from those in the output layer of the network. The argument of the activation function of each hidden unit in an RBF network computes the Euclidean norm (distance) between the input vector and the center of that unit. Meanwhile, the activation function of each hidden unit in a BPN computes the inner product of the input vector and the synaptic weight vector of that unit.

iii. The hidden layer of a RBF network is nonlinear, whereas its output layer is linear. However, the hidden and output layers of a BPN are usually all nonlinear.

iv. BPN constructs global approximations to non-linear input-output mappings. On the other hand, RBF networks using exponentially
decaying localized nonlinearities (e.g., Gaussian functions) construct local approximations to nonlinear input-output mappings.

v. The usual multilayer perceptron builds its classification from hyperplanes, defined by the weighted sums $\sum_j w_{ij} x_i$ which are arguments to nonlinear functions, whereas the radial basis approach uses hyperellipsoids to partition the pattern space. These are defined by the functions of the form $\phi(\|x - y\|)$ where $\|\ldots\|$ denotes some distance measure. The advantage of using the radial basis approach is that once the radial basis functions have been chosen, all that is left to determine are the coefficients of the RBF to allow them to partition the space correctly. Since these coefficients are added in a linear fashion, the network provides a guaranteed solution since there is no local minima situations in which to fall. In effect, the radial basis functions have expanded the inputs into a higher-dimensional space. This approach is guaranteed to produce a function that fits all the data points, as long as there is a function for input to be separated.

vi. In the absence of any knowledge about the data, the radial basis functions are chosen so that they fit points evenly distributed through the set of possible inputs. The use of radial basis functions is becoming more popular, since they need only linear optimization techniques, which provide a guaranteed and globally optimal solution.

vii. BPN network training can result in providing weights in undesirable local minima of the criterion function and because of their highly nonlinear structure, and it gets worse as the network size increases. This difficulty has motivated many researchers to search for a structure where the output depends on the network weights is less nonlinear. The RBF network has a linear dependence on the output layer weights, and
the nonlinearity is introduced only by the cost function for training, which helps to avoid local minima. Additionally, this network is inherently well suited for blind signal processing because it uses unsupervised learning to cluster the input data. Since the local response power of RBF networks offers great classification and approximation capabilities, the Gaussian RBF network is used as a good function approximator.

In addition to above, RBF network also has the following characteristics.

(i) It has faster learning capability.

(ii) It finds the input to output map using local approximators. Usually the supervised segment is simply linear combination of approximators. Since linear combiners have few weights, these networks train extremely fast and require fewer training samples.

(iii) Faster convergence, smaller interpolation errors, higher reliability and a more well–developed theoretical analysis than BPN.