Analysis of Independent Component Analysis and Nonnegative Matrix Factorization

Anisa Aisha Ahmed
CS, Dhanani School of Science and Engineering
Habib University
aa04049@st.habib.edu.pk

Hiba Jamal
CS, Dhanani School of Science and Engineering
Habib University
hj04033@st.habib.edu.pk

***Abstract*—Representation and decomposition of multivariate data is a fundamental problem in machine learning, specifically neural network research. Employing linear transformation techniques on multivariate data is a common approach to working with multivariate data. For this purpose, various approaches have been developed. This study explores two of these approaches; Independent Component Analysis and Non-negative Matrix Factorization. We choose the application of signal source separation. We conduct different scenario based experiments and conclude that where NMF works very well as a decomposition algorithm, ICA is the better choice to predict the source signals from a mixed input signal set. The complexity of these convergence-based algorithms is also calculated to provide a better overview of computational performance.**

I. Introduction

Cocktail party problem is a famous auditory signal analysis problem. It has been treated as a benchmark problem for blind source signal separation problem where the goal is to focus on the underlying source signals from a mixture of signals. In other words, here we would have a sample of multivariate data (mixture of different audio signals), that need to be decomposed into original data.
A decomposition from original data to source data is thus required. ICA and NMF are two algorithms that fit this purpose. They are capable of taking in input multivariate data, and perform a decomposition on it to regain the source data (or signals in our example).

II. Literature review

Multivariate data analysis and decomposition is a popular research area in machine learning. Linear Discriminant Analysis (LDA), Singular Value Decomposition (SVD), Cholesky decomposition and Principal Component Analysis (PCA) are some of the methods developed for this purpose.
Both ICA and NMF follow a standard method, however, both these algorithms utilize functions that may be changed as hyperparameters. In ICA, this function is used to introduce non-linearity, improving the accuracy and speeding up the convergence. The most common function used is the tan hyperbolic [2]. Xu et al. also describe the use of a novel non-linear function using sine function in [5].

In NMF, the function that may be changed is the optimization function used to minimize the approximation. Frobenius norm or Kullback-divergence are common optimization methods as discussed in [6][4]. A probabilistic approach to this algorithm has also been developed as described by Ito and Amagasa [3].

III. Mathematical Background

A. Independent Component Analysis

Independent component analysis is a supervised learning algorithm, which works under the assumption that the input vectors are independent of each other, and have non-Gaussian distributions. These assumptions are important as the goal of the algorithm is to disentangle independent information from given multivariate data. Without the input vectors being independent, this will not be possible.
ICA requires two pre-processing steps; centering and whitening of input vector. This is done to simplify the ICA algorithm. Centering is performed by subtracting the input vector by its mean vector. In whitening, the vector is transformed linearly so that the correlation between its components is removed. This is done using eigenvalue decomposition of the covariance matrix. Mathematically,

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

where D is the diagonal matrix of eigenvalues, and E is an orthogonal matrix of eigenvectors.
After the preprocessing, the ICA algorithm proceeds by initializing a weight vector W, and minimizing the Gaussianity of projection $W^T x$. A non-quadratic function is then used to speedup the convergence and improve the accuracy. For our purposes, we'll assume this non-quadratic function to be the tan hyperbolic.

$$g(x) = \tanh(x)$$

$$g'(x) = 1 - \tanh^2(x)$$

Each component in the weight vector is then updated until convergence. A summary of the algorithm is found in the pseudocode below.

Algorithm 1: Independent Component Analysis

Result: Input vector x
Initialize W with random values;
for W_p *in* W **do**
  $W_p = \frac{1}{n}\sum_i^n Xg(W^TX) - \frac{1}{n}\sum_i^n g'(W^{TX})W$;
  $W_p = W_p - \sum_{j=1}^{p-1}(W_p^TW_j)W_j$;
  Normalize W_p;
  Repeat steps until convergence;
end

B. Non-negative Matrix Factorization

Non-negative matrix factorization creates a decomposition of a given non-negative input matrix V, into two non-negative matrix factors W and H. This factorization is an approximation of the input matrix V, and hence will be represented as,

$$V \approx WH$$

This decomposition is significant because the matrices W and H represent useful information. The matrix W represents the basis vectors of input matrix V, and the matrix H represents the weights or activations. In other words, each data sample in matrix V is represented by a linear combination of the basis vectors in V which is encoded by the weights in matrix H.
Since this factorization is an approximation of V, the quality of this approximation can be improved by using different optimization functions to minimize the distance between WH and V (as mentioned in Section 2). Common optimization functions used for non-negative factorization are Frobenius norm and Kullback-Leibler divergence.
Among the different approaches of optimizations functions, we will use the Kullback-Leibler divergence method in this project. This is mainly because it is the most common optimization function for this problem. The cost function for Kullback-Leibler divergence is given by,

$$D(V||WH) = \sum_{ij} V_{ij}\log(\frac{V_{ij}}{[WH]}_{ij} - V_{ij} + [WH]_{ij}$$

A completely accurate decomposition of matrix V can be obtained if the optimization function is minimized to zero, that is,

$$D(V||WH) = 0$$

Optimizing this function is tricky because this function is convex in either W or H but not for both W and H. To deal with this, iterative algorithms are used to minimize the function. For the minimization, multiplicative update rules developed by Lee and Seung [4] have been proved to provide results with good convergence.

$$W \leftarrow W \cdot \frac{\frac{V}{WH}H^T}{H^T}$$

$$H \leftarrow H \cdot \frac{W^T\frac{V}{WH}}{W^T}$$

In summary, the algorithm can be described as,

Algorithm 2: Non-negative Matrix Factorization

Result: Matrix V
initialize W and H with random values;
while *convergence* **do**
  update W using multiplicative rule for W;
  update H using multiplicative rule for H;
end

IV. Theoretical Analysis

Both the algorithms were analyzed in terms of theoretical complexity to determine their efficiency. Here we analyze the time complexity for both the algorithms, and their pre-processing steps.

A. Independent Component Analysis

The time complexity for pre-processing steps (centering and whitening) in ICA is $\mathcal{O}(n) + \mathcal{O}(n^3)$. The centering operation calculates the mean along one axis in n steps. During the whitening operation, the eigenvalue decomposition step is expensive which takes n^3 steps to execute.
For the algorithm, ICA performs operations on N number of samples, and executes i iterations until convergence. Within these i iterations, new value of weight matrix W is computed according to the update rules in Algorithm 1. This brings the total complexity of the algorithm (without pre-processing step complexity) to be $\mathcal{O}(N*(n*f+n+f^2))$, where f is the number of features extracted from the vector X. this brings the complexity of the algorithm to be $\mathcal{O}(n^2 * f)$ Since there is uncertainty in the convergence of the algorithm, we assume a relative error E, and add the rate of convergence to the final complexity as $\mathcal{O}(n^2 * f\log(E))$.

B. Non-negative Factorization

Time complexity of NMF largely depends on the type of optimization function used. In this project, time complexity of NMF is calculated with regards to the KL-Divergence method used. NMF performs operations n number of maximum iterations. As described in Algorithm 2, the matrices W and H are updated using matrix multiplication which takes $\mathcal{O}(\mathcal{FKT}$ steps, where F =columns of input matrix V, T =rows of input matrix V, and K =rows in the decomposition matrix W or columns in decomposition matrix H. Similar to ICA, time complexity of NMF also depends on the rate of convergence. Here also we assume a relative error E, and add it to the final complexity as $\mathcal{O}(FKT\log(E))$.

V. Experiments

The comparison between both algorithms was established by considering a common application for both problems; blind source separation. In this regard, two experiments were performed on both the algorithms.

1) The input was a mixture of 3 different noise induced signals. This simulates the cocktail party problem but with multiple recording devices set in a room at different locations to capture a variety of intensities of the sound

being recorded. The decomposition should result in the three original signals. It is shown in following figure.

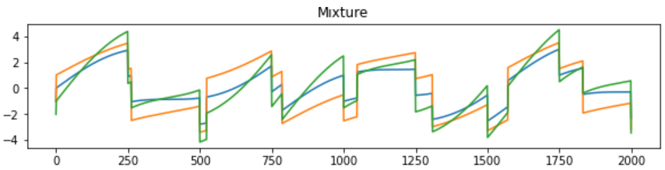

Figure 1. Mixture of 3 different noise induced signals.

2) The input signals were added to make a single signal. This simulates the cocktail party problem but with a single recording device so each sound source is mixed into a single input signal. As shown in Fig. 2. The decomposition should result in the three original signals.

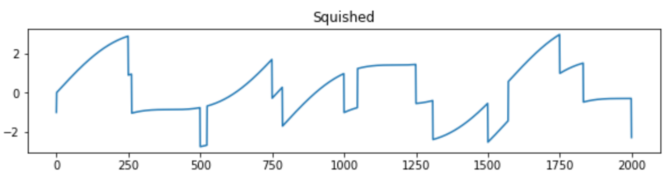

Figure 2. Input signals added into a single signal.

Running ICA on the mixture of signals as shown in Fig. 1, we saw that prior convergence (to max. iterations) was not achieved for all three of the decomposed signals. The maximum number of iterations were set to 10000. We see that the first and second generated signal reach convergence much before than the third signal, which keeps running till the last iteration in Fig. 3.

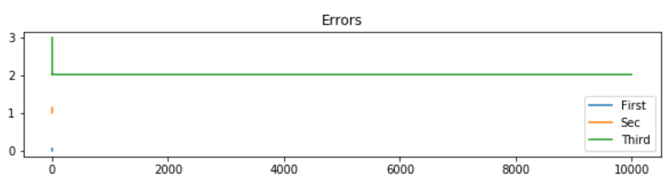

Figure 3. Convergence points of the three signals. Convergence is reached when the new W matrix is almost equal to the previous W matrix.

The success of ICA to generate the original sources is shown in Fig. 4.

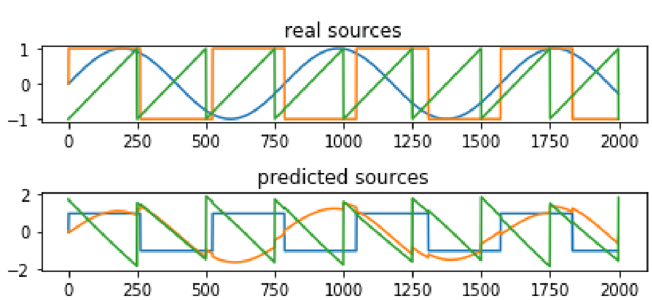

Figure 4. ICA decomposition results.

Next, running the ICA algorithm on the single signal as shown in Fig. 2 to simulate the second scenario gave the following results. The convergence was reached for only the first signal, the other two were running till 50000 iterations, as shown in Fig. 5.

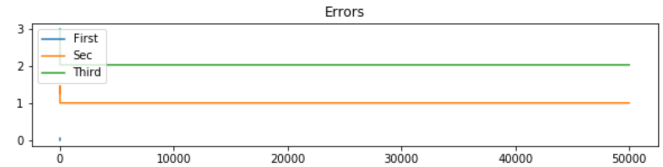

Figure 5. Convergence points of the three signals. Convergence is reached when the new W matrix is almost equal to the previous W matrix.

Whereas, in running the NMF algorithm with the value of $K = 3$ directly on the mixture of input signals we observed that it gives back the original mixture of signals. This is because the convergence criteria for NMF is the minimization of the distance between the decomposed matrices WH and V, which in this case is the mixture of signals as shown in 1. This minimization can be seen in Fig. 6. Nonetheless, we see the approximation is quite good, as shown in Fig. 8. Hence, it is evident that we will have to add extra steps to attain the original sources.

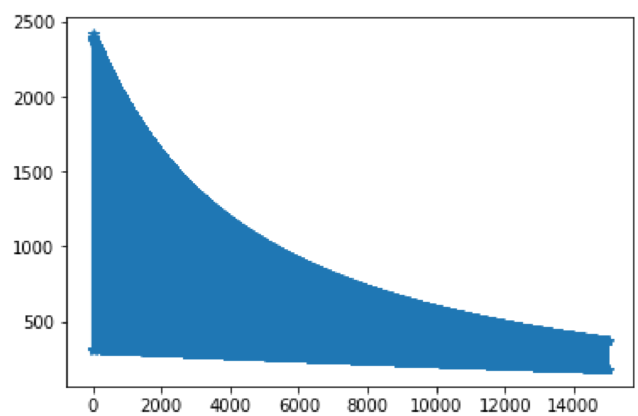

Figure 6. Distance decreasing between the original matrix V of mixed signals and approximation WH till 15000 iterations.

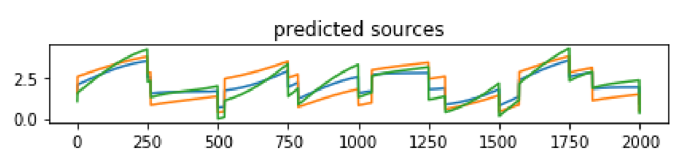

Figure 7. Prediction of Fig. 1 by the NMF algorithm.

To determine the different phase components of the signal, Short Term Fourier Transform [1] was applied to the input signal (as shown in Fig. 2). This was scaled to all positive values and passed as input to the NMF algorithm with the value of K set as 3, since that is how many sources we have.

The minimization is distance between the STFT input and NMF output is shown in Fig. 8.

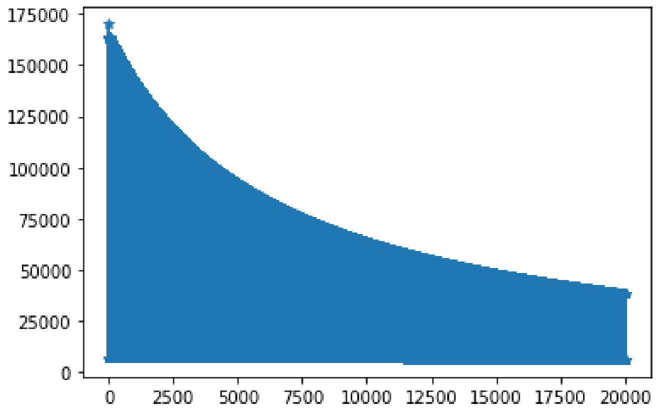

Figure 8. Distance decreasing between the original matrix V of added signals and approximation WH till 20000 iterations.

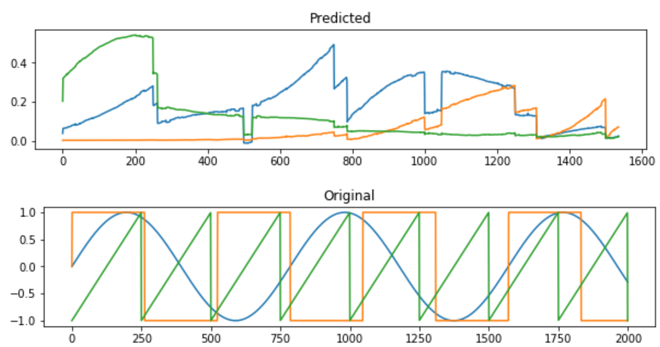

Figure 9. NMF for decomposition of added signals.

VI. Results

Nonetheless, the results tell us that ICA is a much better choice for this specific example. The NMF approximation of source signals is given in Fig. 9. This was achieved after taking the inverse STFT of the matrix product of individual basis vectors in W and their weight encodings in H. This difference in results from ICA could be attributed to the loss of information during the STFT and inverse STFT operations. Nonetheless, it does try to replicate the nature of source signals to a certain extent.

References

[1] Mariam Hakobyan. "Supervised sound source separation using Nonnegative Matrix Factorization". PhD thesis. July 2018. DOI: 10.13140/RG.2.2.27340.72326.

[2] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". In: *Neural Networks* 13 (2000), pp. 411–430.

[3] Hiroyoshi Ito and Toshiyuki Amagasa. "An Optimization Scheme for Non-Negative Matrix Factorization Under Probability Constraints". In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8. DOI: 10.1109/BIGCOMP.2019.8679467.

[4] Daniel D. Lee and H. Sebastian Seung. "Algorithms for Non-Negative Matrix Factorization". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. NIPS'00. Denver, CO: MIT Press, 2000, pp. 535–541.

[5] Pengfei Xu, Yinjie Jia, and Zhijian Wang. *Blind source separation using Fast-ICA with a novel nonlinear function*. 2019. arXiv: 1907.03432 [eess.SP].

[6] Zhirong Yang et al. "Kullback-Leibler Divergence for Nonnegative Matrix Factorization". In: vol. 6791. June 2011, pp. 250–257. ISBN: 978-3-642-21734-0. DOI: 10.1007/978-3-642-21735-7_31.