ICA vs NMF vs PCA for BSS: Independent Component Analysis (ICA) versus Non-negative Matrix Factorization (NMF) versus Principal Component Analysis (PCA) for Blind Signal Separation (BSS) of both audio and images

Marcos Cantu
Boston University
Boston, MA 02215
cantu@bu.edu

Michael Clifford
Boston University
Boston, MA 02215
jcliff@bu.edu

Andrew Novicki
Boston University
Boston, MA 02215
novicki@bu.edu

Austin Welch
Boston University
Boston, MA 02215
austinmw@bu.edu

Abstract

We implemented several approaches to Blind Signal Separation (BSS) of both audio and images. While Principal Component Analysis (PCA) is not well-suited for BSS, Independent Component Analysis (ICA) achieves impressive signal separation results for linear time-invariant mixtures of concurrent talkers. However, it is incapable of source separation for the time-variant case; this shortcoming was demonstrated using a computational model of room acoustics and modifying the room impulse responses to be either time-variant or time-invariant. NMF was not effective for source separation of speech, as it instead separated the mixture into low frequency vowels and high frequency sibilants. For image separation, all three methods were able to distinctly separate original images from the mixed input; however, there were very mixed results. Images with strong colors and no large solid colored backgrounds were reconstructed accurately. Images with solid colors in the background typically had a "ghost image" *of one of the other images in the background. Quantitatively, the results across the two NMF methods and* FastICA *were comparable, though the divergence method of NMF was the most successful across all metrics.*

1. Introduction

Our course project consisted of a comparison between three distinct approaches to the *Blind Signal Separation* (BSS) problem for both audio signals and images. While the algorithm we learned in class, *Principal Component Analysis* (PCA) [27, 34] was included in the comparison, It was not expected to perform as well (Fig. 1) as the other two algorithms: *Independent Component Analysis* (ICA) and *Non-negative Matrix Factorization* (NMF).

While the ICA and NMF algorithms are presented here within the context of the field of *Machine Learning* [7, 16, 38], both algorithms first gained traction within the field of *computational neuroscience* [23, 31]. This presents a unique opportunity to explore a topic in machine learning that relates to auditory and visual sensory processing, as well as *Digital Signal Processing* (DSP) [43, 54] of both 1-D signals and images. One of the challenges will be testing the ICA and NMF algorithms on both types of signals.

2. Literature Review

Blind Signal Separation (BSS) [13, 33, 9] defines a broad class of signal processing problems where there are multiple signal sources and multiple sensors (signal receivers) and the signal sources are mixed together in some unknown (blind) fashion. The goal of BSS is to estimate the mixing process so that it can be inverted to generate the original separate signals. *Independent Component Analysis* (ICA) [4, 29, 2, 6, 24, 22, 32, 35, 25] was first put forth as an approach to solving the BSS problem in 1984 [20]. Although ICA for BSS was first discussed in the context of neural circuits and synapses [20, 26, 23], it was more rigorously put forth by Comon in 1994 [12] and gained popularity in both the neural networks and computational neuroscience fields in 1995 when Bell and Sejnowski [5], developed an improved ICA algorithm. In a 1999 *Nature* paper by Seung and Lee [31], *Non-negative Matrix Factorization* (NMF) [11, 36, 14, 31, 52, 51] emerged as an alternative to ICA. Since their emergence in the 1990s, the ICA and NMF algorithms have been applied to numerous problems in machine learning and signal processing. For the project summarized here, we will assess the effectiveness of the NMF and ICA (and PCA) algorithms for solving the BSS problem for both mixtures of audio signals and mixtures of images.

3. Problem Formulation and Solution Approaches

The BSS problem consists of finding a solution to the following linear equation describing a mixing process:

$$x = A \cdot s \qquad \leftrightarrow \qquad s = A^{-1} \cdot x$$

Where x is a vector defined as the output of multiplying the original signal vector s with an unknown mixing matrix A. Can we find an estimate for A^{-1}, effectively reverse the mixing process, and isolate the original signal vector s? This problem is complicated further by the fact that the original signal vector, s, is also unknown.

The ICA approach to BSS posits that each signal $s_i(t)$ is independent of of every other signal $s_j(t)$ in the mixture. Hence, the algorithm relies on a measure of independence. In the *FastICA* [21] algorithm, this is effected through maximizing the kurtosis. The PCA algorithm is ill-suited for recovering the original sources because it finds orthogonal directions of maximum variance whereas the ICA algorithm finds maximum independence.

Within the class of problems broadly defined as BSS, there are generally speaking two types of scenarios that can be modeled depending on the number of signal sources and the number of signal receivers: The determined case and the underdetermined. The latter represent a scenario where there are fewer signal receivers than signals. This is a hard problem and remains an active area of research. For the project at hand we primarily focused our efforts on addressing the former, the determined case, where there are an equal number of signal and receivers, requiring an $n \times n$ unknown mixing matrix to estimate.

3.1. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a well-known model for unsupervised learning of an unmixing matrix A^-1 from the output signals x alone. ICA achieves this approximation by maximizing the independence between the observed signals. There are a number of variations of ICA found throughout the literature, and three in particular we will discuss later in this paper, but all ICA algorithms share two main elements: A defined metric for measuring independence (such as kurtosis or negentropy) and an optimization step (often either a gradient based method or a fixed point algorithm).

Let A be the square ($n \times n$) unknown mixing matrix used to generate our output signals $x = \{x_1, ..., x_n\}$ from our input signals $s = \{s_1, ..., s_n\}$ such that $As = x$. The goal of ICA is to estimate an unmixing matrix W, where $W = A^{-1}$ such that $s_i = Wx_i$ so that we can recover our original signals s. Let the probability density function (PDF) of $s = P_s(s)$. Note from above that:

$$x = As = W^{-1}s \quad \text{and} \quad s = A^{-1}x = Wx$$

We can write the probability function of $x = P_x(x)$ as $P_s(Wx) \cdot \det(W)$. Assuming independence between each of our signals, we can write the ICA model as:

$$P_s(s) = \prod_i^n P_s(s_i)$$

Since we know $s = Wx$, we can write $p(x)$ as:

$$p(x) = \prod_i^n p_s(w_i^T x) \cdot |W|$$

The *log-likelihood* (ℓ) of the unmixing matrix W:

$$\ell(W) = \sum_i^n \Big(\sum_j^n \log g'(w_j^T x^{(i)}) + \log |W| \Big)$$

We will use the following gradient ascent learning rule, which converges to an estimate of the unmixing matrix W:

$$W := W + \alpha \left[\begin{pmatrix} g^{''}(W_1^T x^{(i)}) \\ ... \\ g^{''}(W_n^T x^{(i)}) \end{pmatrix} + (W^T)^{-1} \right]$$

We have now trained our ICA model and can recover our original signal set s, with our learned unmixing matrix W by $s = Wx$. The above derivation represents the Maximum Likelihood Estimation (MSE) approach [41, 48, 49] to ICA.

3.2. Non-negative Matrix Factorization (NMF)

The NMF algorithm decomposes a data matrix Y as a product of two matrices A and X having only non-negative elements [31, 10]. Y is a $n \times m$ matrix, while A is an $n \times k$ matrix and X is a $k \times m$ matrix. The columns of A and the rows of X represent the rank of the mixing matrix, and in using NMF for BSS, it should be equal to the number of mixed images. Lee and Seung [31, 32] describe an iterative algorithm for finding values of A and X that solve the equation $Y = AX$, satisfying the condition that both A and X are nonnegative. They suggest two possible cost functions to minimize to solve for A and X: one representing the distance between Y and AX, and one that represents the divergence of A and B. The distance cost function can be represented as:

$$C = |||Y - AX||^2$$

With the distance cost function, A is adjusted by the multiplicative update equation

$$A_{ia} = A_{ia} \frac{(Y^T)_{ia}}{(AXX^T)_{ia}}$$

X is updated by the multiplicative update equation

$$X_{a\mu} = X_{a\mu} \frac{(A^T Y)_{ia}}{(A^T AX)_{ia}}$$

The divergence cost function can be represented as

$$C = D(Y||AX) = \sum_{i,j} (Y_{ij} log(\frac{Y_{ij}}{(AX)_{ij}}) - Y_{ij} + (AX)_{ij}).$$

A is adjusted by the multiplicative update equation

$$A_{ia} = A_{ia} \frac{\sum_u (X_{au} Y_{i\mu})/(AX)_{i\mu}}{\sum_v X_{av}}.$$

X is updated by the multiplicative update equation

$$X_{a\mu} = X_{a\mu} \frac{\sum_i A_{ia} Y_{i\mu}/(AX)_{i\mu}}{\sum_k A_{ka}}$$

The algorithm iterates until the cost function reaches a minimum, and is guaranteed to reach a local minimum; however, the minimum is not unique [32, 36, 1]. Blind source segmentation of mixed images can be implemented by assuming the mixed images are represented by Y. After running the algorithm, the separated images are extracted as rows of X. One difficulty in using NMF for blind source separation is that the algorithm has a non-convex solution space with many local minima [36, 1].

4. Implementation (Datasets + MATLAB code)

4.1. PCA, ICA and NMF for Audio

Datasets and Evaluation for Audio.
Our original intention was to use datasets from the *2015 Signal Separation Evaluation Campaign* (SISEC) [42] and/or the *Blind Audio Source Separation evaluation database* (BASS-dB) [18]. However, the *BASS-dB* database online repository no longer contains audio files for the determined case and only multi-track (i.e. music) signals were still available on their website. Hence, the decision was made to construct our own mixtures using the *BUG* corpus of speech sounds [28] developed here at the Boston University campus. The *BUG* corpus has 18 unique talkers sampled at 50kHz with .WAV files for individual words. These component words were used to construct the two-word phrases *"Cheap Cards"*, *"Red Toys"*, and *"New Hats"*, each uttered by a different talker. These phrases were used to generate the three-talker mixtures used for the data in Table 1 and Figs. 1-2. Performance of the algorithms was evaluated using the *Source to Distortion Ratio* (SDR) and *Source to Interferences Ratio* (SIR) performance metrics from the *BSS Eval* toolbox [50, 17]. Performance of the algorithms was evaluated in both additive Gaussian noise (Fig. 1) and room acoustics (Fig. 2). For the latter, we used a *room-image* method model of room acoustics [3, 46, 15] where the positions of individual microphones, as well as the positions of multiple sources, can be specified.

PCA audio implementation in MATLAB
The project guidelines suggested that we explore at least one algorithm that is part of the syllabus. For our BSS topic, this was Principal Component Analysis (PCA), which not perform as well as the ICA and NMF algorithms, as it is ill-suited for the BSS problem. We used MATLAB code for PCA AUDIO written by Brian Moore [37].

ICA audio implementation in MATLAB
For ICA approaches to BSS of audio signals, we implemented code from James V. Stone's book [48] for Bell and Sejnowski's ICA algorithm [5] and the original author written code for both Hyvarinen's *FastICA* [21] algorithm and the *Efficient FastICA* (EFICA) [30] algorithm.

NMF audio implementation in MATLAB
We implemented NMF for audio using MATLAB code from a *Source Separation Tutorial* at Stanford University [8]. We used code written by Romain Hennequin [19], who implemented Smaragdis' *Non-Negative Matrix Factor Deconvolution* (NMFD) [47]. For Schmidt and Mørup's *Non-Negative Matrix Factor 2-D Deconvolution* (NMF2D) [45], we used the *demo* code [44] written by the authors. NMFD modifies the original NMF algorithm [31] by adding convolution over time while NMF2D adds convolution across both time and frequency. All NMF variants operate on the magnitude of the *Short Time-Fourier Transform* (STFT) [43, 39, 40] and require post-processing *re-synthesis* via an *inverse Short-Time Fourier Transform* (iSTFT).

4.2. ICA and NMF for Images

Datasets and Evaluation for Images. A lack of multi-channel mixed image data necessitated generating data for testing. This was achieved with the following steps: First, three images of equal dimensions were required. Each 3-D RGB image matrix was reshaped into a one-dimensional vector and stacked on top of each other. A square mixing matrix with dimensions equal to the number of original images was generated with a pseudorandom standard uniform distribution generator on the open interval (0,1). This mixing matrix was multiplied by the stacked image matrix to obtain a mixed matrix. Each row of this mixed matrix represents a single channel mixed image, and the row can be extracted and reshaped into a three-dimensional RGB image matrix for viewing. Given PCA's poor performance on BSS of audio signals (see Table 1 and Figs. 1-2), the decision was made to focus on the NMF and ICA algorithms for blind source separation of image mixtures.

ICA images implementation in MATLAB
While the original author written code for *FastICA* [21] was used for the results on audio, team member Austin Welch wrote his own MATLAB function for implementing *FastICA* for mixtures of images. The FastICA algorithm requires two preprocessing steps. The first is centering the data. The second is called whitening the data. It requires performing eigendecomposition of the covariance matrix of the centered data in order to obtain an eigenvector matrix $\mathbf{E}$ and a diagonal eigenvalue matrix $\mathbf{D}$. After these are calculated, the prewhitening transformation is defined as: $\mathbf{ED}^{-\frac{1}{2}}\mathbf{E}^{\mathbf{T}}\mathbf{X} \rightarrow \mathbf{X}_{\mathbf{prewhitened}}$. Once this is completed the FastICA algorithm can be applied to the prewhitened data.

There are many choices of nonquadratic nonlinearity functions that work well with FastICA, but Hyvarinen's paper [21] suggests robust performance with the following following function along with its first and second derivatives: $f(u) = -e^{-\frac{u^2}{2}}$, $g(u) = ue^{-\frac{u^2}{2}}$, $g'(u) = (1-u^2)e^{-\frac{u^2}{2}}$.

Multiple versions of the FastICA algorithm exist, but the one chosen to be implemented is referred to as a multiple component extraction method. It is a fixed-point iteration scheme, which seeks an orthogonal rotation of the prewhitened data in order to maximize non-Gaussianity of the individual components. FastICA was applied in the context of a determined BSS problem consisting of three mixed-signal images, so the number of desired components, $\mathbf{C}$, is equal to the columnwise dimension of the prewhitened data, $\mathbf{N}$. The horizontal dimension of the data ($\mathbf{M}$) represents the number of pixels in each image. The entire algorithm is as follows:

for p **in** 1 to C:
  $\mathbf{w}_p \leftarrow$ Random vector of length N
  while $\mathbf{w}_p$ changes
    $\mathbf{w}_p \leftarrow \frac{1}{M}\mathbf{X}g(\mathbf{w}_p^T\mathbf{X})^T - \frac{1}{M}g'(\mathbf{w}_p^T\mathbf{X})\mathbf{1}\mathbf{w}_p$
    $\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1}\mathbf{w}_p^T\mathbf{w}_j\mathbf{w}_j$
    $\mathbf{w}_p \leftarrow \frac{\mathbf{w}_p}{\|\mathbf{w}_p\|}$

Output: $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_C]$
Output: $\mathbf{S} = \mathbf{W}^T\mathbf{X}$

Finally, the output, $\mathbf{S}$, represents the unmixed matrix. Each row can be reshaped to form an individual separated image.

NMF images implementation in MATLAB
The distance and divergence cost functions described by Lee and Seung were implemented in MATLAB [31, 32]. The cost functions are minimized by alternating updates of A and X by the corresponding update rule described previously until the cost function is minimized. The distance method converged much faster than the divergence method, but usually at slightly reduced performance.

5. Experimental Results

5.1. Results for ICA, NMF and PCA audio

Our results demonstrate that all ICA algorithm variants had performance far superior to PCA and all NMF algorithm variants for linear time-invariant mixtures of three talkers (Table 1). For our evaluation of performance in the presence of additive Gaussian noise (Fig. 1), the chronology of ICA algorithm development was visible in the results, with Bell & Sejnowski's 1995 *InfoMax* [5] / *Maximum Likelihood* (ML) [41, 48, 49] algorithm providing better performance than PCA, albeit only at positive SNR levels, and the subsequently developed *FastICA* (1999) and *EFICA* (2006) algorithms achieving successively superior performance.

However, the impressive performance of all three ICA algorithms (ML ICA, *FastICA* and EFICA) offered no improvement over the PCA algorithm when evaluated with a *time-variant* model of room acoustics (Fig. 2, Top panel). Through an entirely artificial manipulation only possible in a computational model of room acoustics, the room impulse responses were time aligned, thereby creating a *time-invariant* model of room acoustics in which the ICA algorithms were able to perform blind source separation, although only at very high *Absorption Coefficient* (AC) values corresponding to very low levels of room reverberation (Fig. 2, Bottom panel). Appendix Figures A7 and A8 demonstrate that the Absorption Coefficient needs to be > 0.995 (i.e. the walls absorb $> 99.5\%$ of sound wave energy) for *FastICA* or *EFICA* to provide > 10dB source separation.

Whereas *FastICA* is able to return the true source waveforms, albeit only for linear time-invariant mixtures, the NMF, NMFD and NMFD2 algorithms all decomposed the *Time-Frequency* representation (i.e. spectrogram) into three components: the first is effectively a *sibilant* component (the high frequency hissing component of "s" and "t"), while the other two are vowels (see Appendix figures A9-A11). This is an interesting and potentially useful (for denoising) result, but performs poorly on the source separation metrics (see Table 1 and Appendix Figures A13-A15).

Algorithm	SDR(dB)	SIR(dB)	SAR(dB)
EFICA	35.64	35.64	235.55
FastICA	26.18	26.17	236.99
ICA	5.51	5.51	230.50
PCA	1.29	1.29	94.00
NMF	-2.66	0.22	3.85
NMFD	-4.84	1.81	-0.09
NMF2D	0.54	3.58	5.22

Table 1. Results for ICA, PCA and NMF algorithms for blind source separation of a linear mixture of three concurrent talkers. EFICA was the most successful overall while all NMF variants performed poorly relative to the ICA and PCA algorithms.

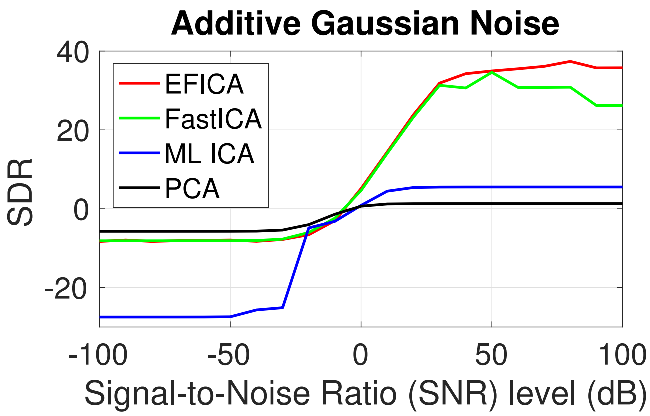

Fig. 1. BSS audio results for additive Gaussian noise. ML ICA outperforms PCA, but only at positive SNR levels. EFICA (red) had the overall best performance but only outperformed *FastICA* at high ($>$ 25dB) SNR levels.

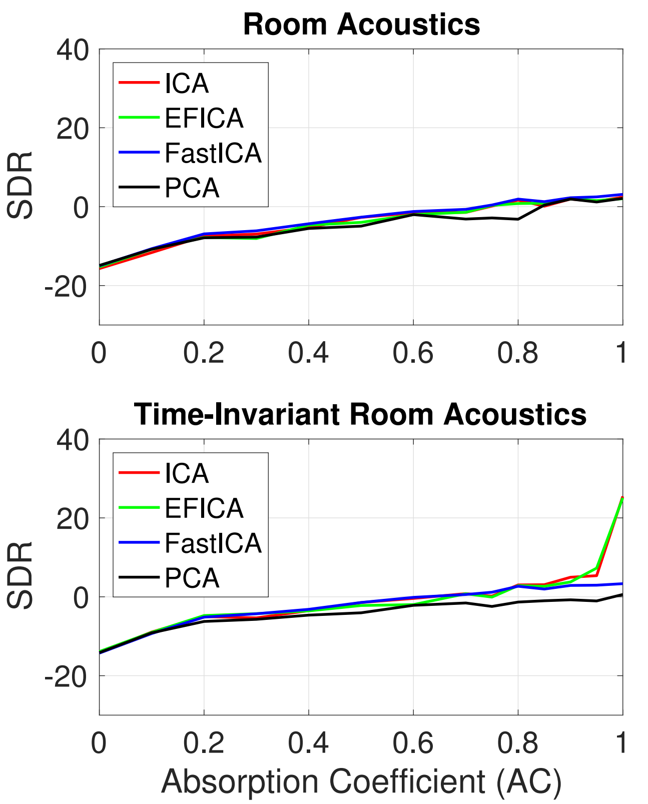

Fig. 2. BSS audio results for a model of room acoustics. ICA algorithms cannot effect blind source separation for a *time-variant* mixture of three talkers in a *room-image* method model of room acoustics (top panel). When the room impulse responses were time aligned (bottom panel), thereby creating an artifical *time-invariant* model of room acoustics, the ICA algorithms were able to perform blind source separation, albeit only at very low levels of reverberation; i.e. only at high *Absorption Coefficient* (AC) values.

5.2. Results for ICA and NMF images

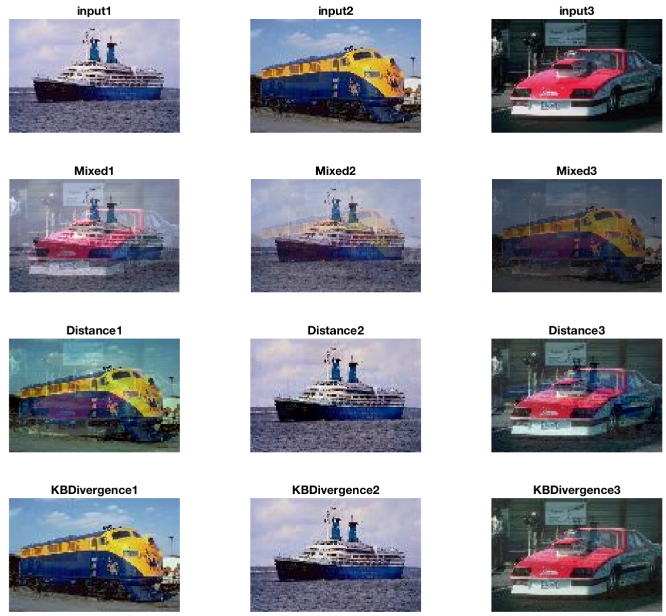

Fig. 3. "Good" NMF results for mixtures of 3 images. Images with strong colors and no large solid colored backgrounds are usually able to be reconstructed accurately.

Fig. 4. "Bad" NMF results for mixtures of 3 images. Images with solid colors in the background typically had an interfering *"ghost image"* in the background.

A test of the image separation was done by separating 3 test images from the COREL Image Database. Input images from three random classes were selected and mixed into three mixed images by a uniform random distribution, BSS was then performed on the three images to separate into the original images.

Qualitatively, both the *FastICA* and NMF algorithms are able to extract the images, but with mixed results (see Table 2 and Figs. 3-6). The separated images were evaluated over a range of metrics using the *BSS Eval* toolbox [17]. In addition, the structured similarity index (SSIM) was also calculated between the original unmixed image and the separated image [53]. The metrics were computed across FastICA and the distance and divergence methods of NMF.

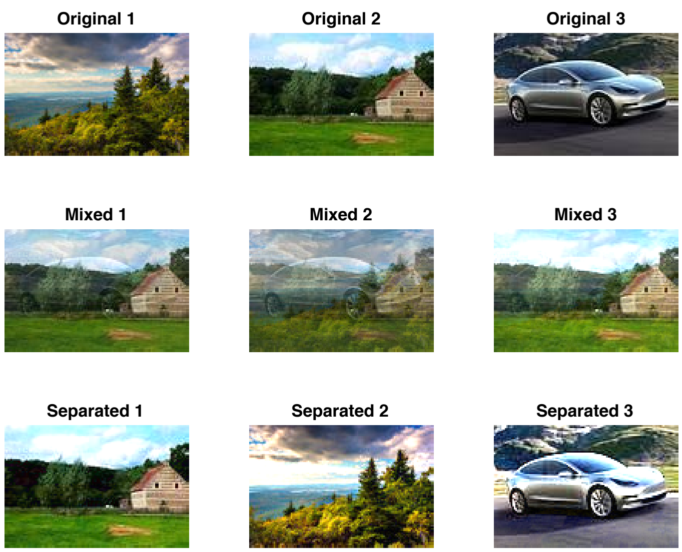

Fig. 5. "Good" ICA results for mixtures of 3 images. Images with strong colors and no large solid colored backgrounds are usually able to be reconstructed accurately.

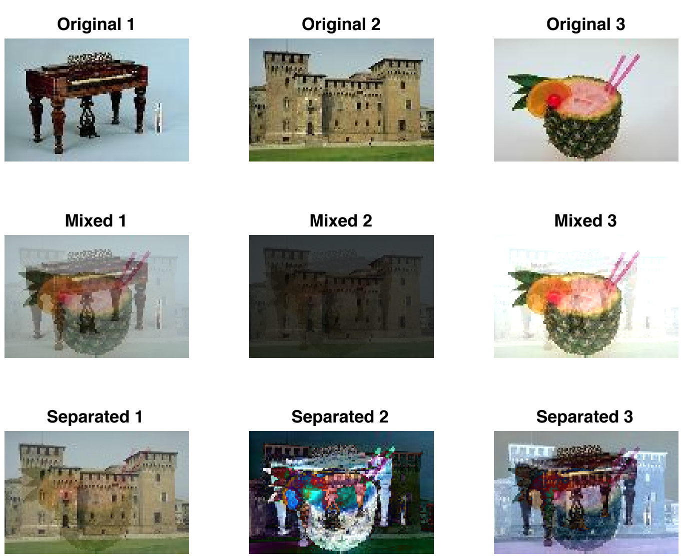

Fig. 6. "Bad" ICA results for mixtures of 3 images. Images with solid colors in the background typically had an interfering *"ghost image"* in the background.

	NMF Div	NMF Dist	*FastICA*
SSIM	0.765 ± 0.05	10.658 ± 0.063	0.603 ± 0.076
SDR (dB)	14.03 ± 2.81	10.33 ± 1.99	7.28 ± 1.00
ISR (dB)	15.33 ± 2.80	11.70 ± 2.17	8.55 ± 0.841
SIR (dB)	22.06 ± 3.03	17.92 ± 1.24	17.37 ± 2.19
SAR (dB)	31.19 ± 3.76	26.83 ± 3.77	23.89 ± 2.48

Table 2. Results for the *FastICA*, NMF Distance (Dist) and NMF Divergence (Div) algorithms for blind source separation (BSS) of a mixture of three images. The mean $\pm$ SEM results were calculated over all the test data. The NMF Divergence (Div) algorithm was the most successful across all metrics

A second test was run on an actual image taken through a window, with the reflection of the window superimposed on the exterior of the window. Images were taken a three different exposure times to represent three mixed images to be separated. The resulting images were unsuccessful at separating the interior from the exterior. A possible reason why the images were not able to be separated is because the different exposures didn't correctly simulate random mixing between the interior and the exterior. The best way to properly simulate the mixing would be to vary the amount of light in the exterior of the window, possibly by taking images at different times of the day.

6. Conclusion

We implemented several approaches to BSSof both audio and images. While PCA is fundamentally poorly suited for BSS, ICA demonstrated impressive signal separation results (see Table 1 and Appendix Fig. A2) for linear time-invariant mixtures of concurrent talkers. The *FastICA* and *Efficient FastICA* (EFICA) algorithms also demonstrated robust performance in additive Gaussian noise (see Fig. 1) at high (> 25 dB) SNR levels. However, all ICA algorithm variants were incapable blind source separation in a time-variant system (Fig. 2, Top panel); this remarkable shortcoming was demonstrated using a computational model of room acoustics and modifying the room impulse responses to be either time variant (Fig. 2, Top Panel) or time-invariant (Fig. 2, Bottom Panel). NMF was not effective for source separation of speech (see Table 1 and Appendix Figs. A13-A15), as it separated the mixture into low frequency vowels and high frequency sibilants (see Appendix Figs. A9-A11).

For image separation, all three methods were able to distinctly separate original images from the mixed input; however, there were very mixed results. Images with strong colors and no large solid colored backgrounds are usually able to be reconstructed accurately (Figs. 3 and 5). Images with solid colors in the background usually had a *"ghost image"* of one of the interfering images in the background (Figs. 4 and 6). Quantitatively, the results across the two NMF methods and *FastICA* were comparable, though the divergence method of NMF was the most successful.

7. Description of Individual Effort

We had two sub-teams implementing the BSS algorithms for use with audio signals and images, respectively. Subteam 1 (Marcos Cantu and Michael Clifford) worked primarily on Section 4.1 (PCA, ICA and NMF for Audio) while Subteam 2 (Andrew Novicki and Austin Welch) focused on Section 4.2 (PCA, ICA and NMF for Images). Marcos Cantu wrote the Abstract, "*1.Introduction*" and "*2. Literature Review*" sections. Michael Clifford wrote the "*3. Problem Formulation and Solution Approaches*" section. The "*3.1 Independent Component Analysis*" section was written by Michael Clifford and Austin Welch. Andrew Novicki wrote the "*3.2. Non-negative Matrix Factorization (NMF)* section. Marcos Cantu wrote the "*4.1 PCA, ICA and NMF for Audio*" section. Andrew Novicki and Austin Welch wrote the "*4.1 ICA and NMF for Images* section". Marcos Cantu wrote the "*5.1 Results for PCA, ICA and NMF audio*" section. Andrew Novicki and Austin Welch wrote the "*5.2 Results for NMF and ICA Images*" section. The *"6. Conclusion"* section was written by Andrew Novicki and Marcos Cantu. All MATLAB scripts for generating Table 1, FIgure 1, Figure 2 and appendix figures A1-A15 were written by Marcos Cantu. MATLAB scripts for generating Table 2 and figures 3 and 4 were written by Andrew Novicki. MATLAB scripts for generating figures 5 and 6 were written by Austin Welch. Marcos Cantu typeset the Final Report and Appendix in LaTeX.

References

[1] Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155 – 173, 2007.

[2] T. Adali, C. Jutten, J. Marcos, T. Romano, and A. K. Barros, editors. *Independent Component Analysis and Signal Separation*. Springer, 2009.

[3] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[4] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 757–763. MIT Press, 1996.

[5] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[6] E. Bingham, S. Kaski, J. Laaksonen, and J. Lampinen, editors. *Advances in Independent Component Analysis and Learning Machines*. Elsevier, 2015.

[7] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] N. Bryan, D. Sun, and E. Cho. Single-channel source separation tutorial mini-series. *Center for Computer Research in Music and Acoustics (CCRMA), Stanford University*, 2013.

[9] A. Cichocki and S.-i. Amari. *Adaptive Blind Signal and Image Processing*. Wiley, 2002.

[10] A. Cichocki, R. Zdunek, and S. Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, May 2006.

[11] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari. *Nonnegative Matrix and Tensor Factorization*. Wiley, 2009.

[12] P. Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314, Apr. 1994.

[13] P. Comon and C. Jutten. *Handbook of Blind Source Separation*. Academic Press, 2010.

[14] A. M. Darsono, T. C. Chuan, M. S. Saat, and A. A. Md Isa. A study of blind source separation using nonnegative matrix factorization. *ARPN Journal of Engineering and Applied Sciences*, 11(18):10702–10708, 2016.

[15] S. Devore, A. Ihlefeld, K. Hancock, B. Shinn-Cunningham, and B. Delgutte. Accurate sound localization in reverberant environments is mediated by robust encoding of spatial cues in the auditory midbrain. *Neuron*, 62(1):123 – 134, 2009.

[16] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Springer, 2000.

[17] C. Févotte, R. Gribonval, and E. Vincent. Bss_eval toolbox user guide–revision 2.0. 2005.

[18] V. Gribonval. Bass-db: the blind audio source separation evaluation database.

[19] R. Hennequin. Matlab code for Paris Smaragdis' NMFD algorithm. *https://github.com/romi1502/NMF-matlab*, 2016.

[20] J. Herault and B. Ans. Circuits neuronaux a synapses modifiables: decodage de messages composites par apprentissage non supervise. *Comptes.-Rendus de lAcademie des Sciences*, 299(III-13):525528, 1984.

[21] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[22] A. Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2012.

[23] A. Hyvärinen and Y. Kano. *Independent Component Analysis for Non-Normal Factor Analysis*, pages 649–656. Springer Japan, Tokyo, 2003.

[24] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000.

[25] A. Hyvärinen and E. Oja. *Independent component analysis: algorithms and applications*. John Wiley & Sons, 2000.

[26] C. Jutten and Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[27] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear PCA criterion in blind source separation: Relations with other approaches. *Neurocomputing*, 22(13):5 – 20, 1998.

[28] G. Kidd, Jr, V. Best, and C. R. Mason. Listening to every other word: examining the strength of linkage variables in forming streams of speech. *J Acoust Soc Am*, 124(6):3793–802, Dec 2008.

[29] I. Koch. *Analysis of Multivariate and High-Dimensional Data*. Cambridge University Press, 2014.

[30] Z. Koldovsky, P. Tichavsky, and E. Oja. Efficient variant of algorithm for fastICA for independent component analysis attaining the cramr-rao lower bound. *IEEE Trans. Neural Net*, pages 1265–1277, 2006.

[31] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[32] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *In NIPS*, pages 556–562. MIT Press, 2000.

[33] A. Liutkus, R. Badeau, and G. Richard. Gaussian processes for underdetermined source separation. *IEEE Transactions on Signal Processing*, 59(7):3155–3167, July 2011.

[34] L. Ma and H. Li. *An Algorithm Based on Nonlinear PCA and Regulation for Blind Source Separation of Convolutive Mixtures*, pages 1–9. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[35] R. Martin-Clemente and S. Hornillo-Mellado. Image processing using ICA: a new perspective. In *MELECON 2006 - 2006 IEEE Mediterranean Electrotechnical Conference*, pages 502–505, May 2006.

[36] A. Mirzal. NMF versus ICA for blind source separation. *Advances in Data Analysis and Classification*, 11(1):25–48, 2017.

[37] B. Moore. PCA and ICA package. *MATLAB Central File Exchange*, 2016.

[38] K. P. Murphy, editor. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[39] S. H. Nawab and T. F. Quatieri. Short-time fourier transform. In J. S. Lim and A. V. Oppenheim, editors, *Advanced Topics in Signal Processing*, chapter 6, pages 289–337. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[40] S. H. Nawab, T. F. Quatieri, and J. S. Lim. Signal reconstruction from the short-time fourier transform magnitude. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '82, Paris, France, May 3-5, 1982*, pages 1046–1048, 1982.

[41] A. Ng. Independent components analysis [course lecture notes for cs229 at stanford university].

[42] N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus. The 2015 signal separation evaluation campaign. In *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation - Volume 9237*, LVA/ICA 2015, pages 387–395, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[43] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.

[44] M. N. Schmidt and M. Mørup. Matlab demo: Non-negative matrix factor 2-D deconvolution for blind single channel source separation, 2006.

[45] M. N. Schmidt and M. Mørup. *Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation*, pages 700–707. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[46] B. G. Shinn-Cunningham, J. G. Desloge, and N. Kopco. Empirical and modeled acoustic transfer functions in a simple room: Effects of distance and direction. *2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 419–423, 2001.

[47] P. Smaragdis. *Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs*, pages 494–499. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[48] J. V. Stone. *Independent Component Analysis: a tutorial introduction*. MIT Press, 2004.

[49] J. V. Stone. Independent component analysis. *Encyclopedia of Statistics in Behavioral Science.*, 2005.

[50] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, July 2006.

[51] D. Wang, R. Vipperla, N. Evans, and T. F. Zheng. Online non-negative convolutive pattern learning for speech signals. *IEEE Transactions on Signal Processing*, 61(1):44–56, Jan 2013.

[52] Y. X. Wang and Y. J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, June 2013.

[53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[54] J. W. Woods. *Multidimensional signal, image, and video processing and coding*. Elsevier/Academic Press, Burlington, MA, 2006.