# NMF techniques for single source audio separation in the BeeMon project


**Student:**

**Andrés Santiago Díez**

**Directed by:**

**Dr. Mitch Parry**

**Dr. Rahman Tashakkori**

Universidad Pública de Navarra

Nafarroako Unibertsitate Publikoa

Appalachian State University

## INDEX

## A- Introduction

### A.1 Case study: the BeeMon project

BeeMon [1] is a project developed by students and staff of the Appalachian State University's Computer Science Department, with the objective of gathering information about the state of monitored beehives with the ultimate goal of helping beekeepers recover the bee populations.

For this purpose, audio processing is performed to detect anomalies in the beehives, and so it is of strong interest to be able to automatically separate the sound generated by the beehive itself from other sources of environmental noise, such as human, animal, or meteorological noise.

With this work it is intended to contribute to the development of the project through tools and information of value to the aforementioned task. In this document 4 interesting methods will be discussed.

## B. Background

### B.1 STFT and spectrograms

It is well known that working in the frequency domain brings many benefits while processing audio, so a good way to represent a recording is combining both time and frequency domains, making a graphical (two-dimensional) image of the audio through spectrograms.

The aforementioned are calculated using the short-time Fourier transform (STFT), which needs to be tuned through the length and overlap of its sliding window. Performing STFT comes with a tradeoff between resolution in both domains, but with a correct choice of its parameters is of very good use for many purposes, and leads to a complex value for every time point and frequency bin, ultimately forming a complex matrix.

These complex matrices computed with STFT are the base for the source separation methods discussed in the following sections of this paper.

### B.2- Audio source separation

One of the the main challenges in the audio processing field could be summarized in being able to improve the quality of certain recordings, separating the desired source of audio from interferent sounds, like all kinds of noise or other captured sources. For this purpose, numerous techniques have been developed over the decades, involving statistical analysis, different kinds of filtering and deep learning, among others.

Audio separation techniques can be classified into informed and blind, depending whether they work with provided training data or without it. In this document the 4 different methods for audio source separation are informed, all four being based on matrix factorizations.

### B.3 MM algorithms

MM algorithms [2] are a type of algorithm based on the iterative optimization of a family of auxiliary functions as a way to optimize a target function. “MM” stands for both minorize-maximize and majorize-minimize algorithms, which means that they can refer to the solution of a problem where either a maximum or a minimum of a certain function is seeked.

In the case of a minimization problem, for it to be suitable, the auxiliary function needs to fulfill certain constraints:

$$\text{For any } \theta: \quad A(\theta|\theta_m) > f(\theta), \qquad A(\theta_m|\theta_m) = f(\theta_m) \tag{1}$$

With these conditions, updating and optimizing the auxiliary function $A(\theta|\theta_m)$ over iteration m will always result in a value of the target function $f(\theta)$ not superior to the one obtained over iteration m-1. Using the convexity of the target function, as m tends to infinity, the algorithm will converge to a local minimum.

The challenge to use these kinds of algorithms is to discover an auxiliary function complying with the mentioned constraints, for which strategies like the Jensen's inequality or the Cauchy-Schwarz inequality are used, among some others. Understanding this was relevant to the study of this project as 2 of the algorithms implemented are MM algorithms.

## C. Non-negative Matrix Factorization

### C.0 Description

Non-negative matrix factorization [3] is a family of algorithms used in the decomposition of matrices in typically 2 matrices whose product is an approximation of the original. This method is used in a wide spectrum of fields, with applications like pattern recognition [4], air emission quality analysis, spectral processing or text mining [5].

These algorithms usually work iteratively updating the calculated matrices, which are nonnegative, to minimize the error between the reconstructed (resulting) matrix and the original one. Intuitively enough, there is always more than one pair of matrices whose product will be a good solution to the problem. Calculating these matrices is an optimization problem that is approached differently by the many types of NMF implementations, like active sets or the gradient descent. In this work NMF through multiplicative update rules [6] is presented, which is the most popular solution due to its efficiency and simplicity of implementation.

### C.1 NMF in audio source separation

Combining what was seen in the previous sections, NMF can be used to factorize an audio spectrogram V into a matrix W of nonnegative spectra and a matrix H of temporal activations [7]. Nevertheless, this itself can not be used directly in audio source separation, as deciding which bases among the matrix W belong to each source is not a trivial task.

Being able to do so would make NMF a good blind source separation method. In reality, NMF based techniques use training data which is decomposed into the aforementioned matrices W and H. The bases generated with this training data can be later fixed as bases for the decomposition of a target audio, in which multiple sources can be separated as the spectral matrix's components' source are known beforehand. An alternative method to this would be to use as basis the STFT of specific small pieces of audio.

Let's use an example to illustrate this; let us have some audio in which the intervention of 2 speakers needs to be separated. Using NMF over a set of small recordings of both speakers alone (training data), a certain amount of spectral bases could be prepared for each of them. The resulting bases are combined into a nonnegative spectral matrix which is used as input to separate the audio with the two speakers. In this new usage of NMF, only the temporal activations' matrix is updated, as the spectral information needs to be fixed to keep track of what bases correspond to each speaker.

The bases that are to be used need to properly model the mixed signal (so that a good reconstruction can be performed) but they also need to be as different as possible between sources, so that the audio corresponding to each of them can be separated well.

Nevertheless, at this point some would notice an important flaw in this method: spectral information is complex and NMF is strictly nonnegative, and so phase information needs to be discarded. In sections E and F a solution to this problem will be addressed.

### C.2 NMF implementation

As said in C.0, the NMF version chosen for this project is based on multiplicative update rules, a method proposed by D. Lee and H. Seung [6]. It seeks the optimization of one of two cost functions based in:

a) Euclidean distance: $\sum_{ft}(V_{ft} - (WH)_{ft})^2$ (2)

b) Divergence: $\sum_{ft}(V_{ij} log \frac{V_{ft}}{(WH)_{ft}} - V_{ft} + (WH)_{ft})$ (3)

The implementation used for this work will use the first option, for which the multiplicative rules are:

$$H \leftarrow H \frac{W^T V}{W^T W H}, W \leftarrow W \frac{V H^T}{W H H^T} \quad (4)$$

A simple python implementation [8] has been used in this work, without major modifications in the original code.

### C.3 Generating bases with NMF

A common usage of NMF in audio processing is performing NMF with factorization rank 1 (using only one base). This methodology provides a spectral vector that well describes the original audio, whenever it is a sound with constant frequency to a certain extent (meaning that its amplitude can vary but its frequency distribution remains somewhat constant along time).

In this work, this method was performed in a database containing over 1350 1-minute pieces of audio sampled at 48kHz that were collected in the beehives studied by the BeeMon project. The errors between the reconstructed single base matrices and the original spectrograms were analyzed, and the best 100 bases were grouped together to be used as bases for further analysis; these bases being the cleanest pieces of audio, as with only one spectral vector is enough to describe the audio over time.

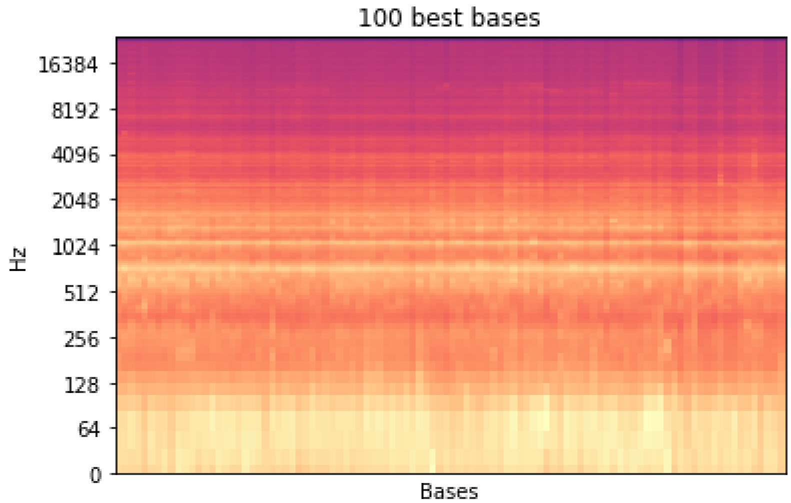



Fig. 1. Spectral representation of the 100 best bases grouped

It can be seen that the best bases, corresponding to the cleanest pieces of audio (as noise would be poorly reconstructed with rank-1 NMF), are similar spectral descriptors of the sound produced by the beehives. Presumably, a number of bases like the one selected may be enough to separate this source of audio due to their similarity. For other purposes like speech separation many more would be needed.

## D- β-NMF

### D.0- Description

β-NMF is a sub-family of algorithms based on the β-divergence [9], which is a divergence estimation method that groups in one expression the Euclidean distance, the Kullback Leibler divergence and the Itakura-Saito divergence. These are famous cost functions commonly used in applications like our case study; the ones mentioned are special cases for $\beta = 2$, 1 and 0 respectively. In the case of the Euclidean distance, it can be classified as a divergence too [10] even if divergences are not usually classifiable as distances. It can be defined as follows [11]:

$$d_\beta(x|y) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)\,y^\beta - \beta\,x\,y^{\beta-1}\right) & \beta \in \mathbb{R}\backslash\{0,1\} \\ x\,\log\frac{x}{y} - x + y & \beta = 1 \\ \frac{x}{y} - \log\frac{x}{y} - 1 & \beta = 0 \end{cases} \tag{5}$$

Since its first and second derivatives are continuous, the same implementation of β-NMF, dependent on the parameter β, can be used for the different cost functions.

**D.1- Implementation**

In [11] multiple algorithms (ME, MM, Heuristic) proposed by previous authors were compared for popular cases of β. In this work a heuristic algorithm based on multiplicative updates (as in the previous section) was implemented for informed audio separation, where only the temporal activations are updated and the bases are predefined and obtained as was explained before.

The Python implementation was based on the NMF code used previously. The updates are performed as follows:

$$H \leftarrow H\frac{W^T[(WH)^{\beta-2}V]}{W^T(WH)^{\beta-1}}\ ,\ W \leftarrow W\frac{[(WH)^{\beta-2}V]H^T}{(WH)^{\beta-1}H^T} \tag{6}$$

In this implementation it can be easily deducted how the updates for the Euclidean distance case (β = 2) are the same as in the previous section.

**D.2- Value comparison**

A set of 100 audios was prepared to compare the efficiency of the algorithm for values β={0, 0.5, 1, 1.5, 2, 2,5} over 100 iterations. In the following figures it is shown the average absolute difference between the original and reconstructed data for a cell, over every iteration.

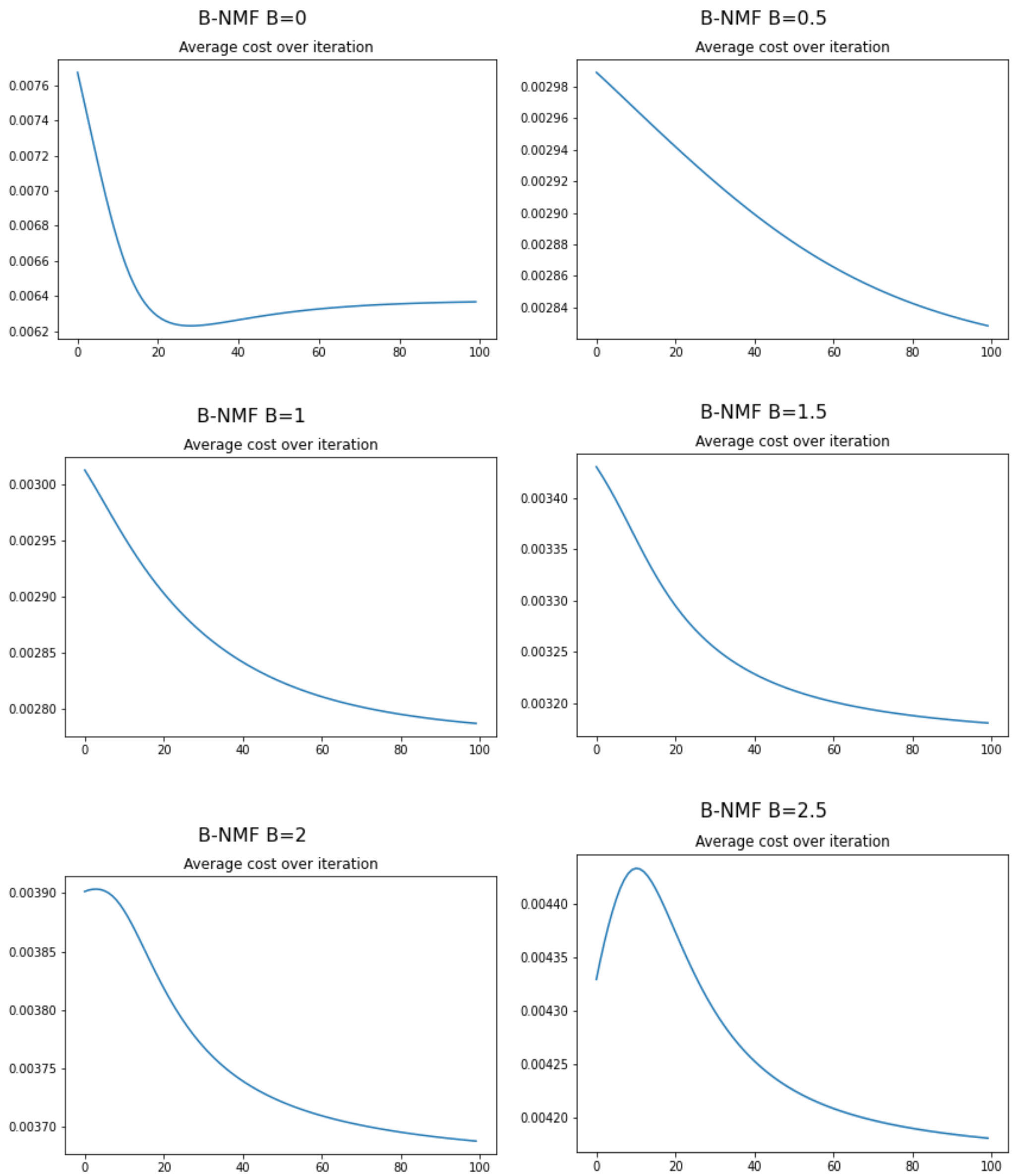



Fig 2. Average cost over iteration for a range of values β={0,1,1.5,2,2.5} while factoring a set of random audios from the BeeMon project.

For the type of audio signals that need to be processed in this project, a value of β=1 proves to work more effectively over 100 iterations, which corresponds to the Kullback Leibler divergence. This value converges to better results than β={0,1.5,2,2.5} and faster to similar results than β=0.5. It may be noted that the Itakura-Saito divergence in this implementation reaches a minimum after 22 iterations and later converges to a higher value and that values greater than β=2 (Euclidean distance) tend to increase the cost over the first iterations.

For simplicity, from now on in this work we will refer as β-NMF to this documented implementation for β=1 and we will refer as NMF to the previous implementation using the Euclidean distance, even if both are particular cases of the β-NMF.

## E- CNMF

### E.0- Introduction to CNMF

As was seen in the previous section, NMF is a method which discards the phase information inherent to the frequency domain when performed in audio processing tasks. Let's consider that NMF is performed to decompose a spectrogram S, that is a mixture of source spectrograms S1 and S2. The addition of the absolute values of its sources does not equal the absolute value of the mixture due to phase cancellation.

$$S = S1 + S2 \qquad \text{but} \qquad |S| \neq |S1| + |S2| \tag{7}$$

The nonnegativity of the method, even when performing at its best, comes with some inherent noise when reconstructing the spectrograms. Apart from that, reconstructing the separated audio with the phase-less calculated spectrograms produces noise too, as it requires using the mixture’s phase information or else reconstructing the phase information with other techniques [12].

As a solution to these problems, a new variant of NMF was proposed, first named Complex NMF [13]. “Nonnegative” was kept in the name because the method updates nonnegative matrices and phase matrices separately. Nevertheless, some posterior publications refer to this algorithm as CMF because the nonnegativity may lead to confusion.

### E.1- Description

The original majorization-minimization algorithm proposed by Kameoka et al. can be found in their paper. As a summary, the algorithm iteratively updates the three necessary components for the reconstruction of the spectrograms, these being the bases with the spectral information, the temporal activations, and the phase information, as well as the auxiliary terms used by the algorithm. However, for informed source separation it is common to fix the spectral bases, so these are not updated by the algorithm.

The same authors later proposed a different algorithm that used the Kullback Leibler divergence [14] instead of the Euclidean distance, which they demonstrated to have slightly better results. Nevertheless, some consider its source separation capabilities to be insufficient, being further variants proposed, including the Phase Constrained CNMF [15] [16], that uses constraints like the phase evolution or phase unwrapping (to better predict the phase of a frequency component belonging to a source over time) or the repetition of audio events.

### E.2- CNMF implementation and problems

In this work a Python implementation of the same author as in the previous section was used, with very slight modifications. The majority of the matricial operations conducted in this program use the Numpy method Einsum. This utility has the flexibility to allow the user to perform linear multi-dimensional operations following the Einstein summation convention as well as other types of operations.

However, CNMF presents a big disadvantage in its usage, which is its overparameterization. Every iteration updates the phase information, which means working with a 3 dimensional matrix of size (F frequency bins * T time points * K bases). In the case of the audios to be processed in this project, using the same spectrograms as in the previous section (K=1025, T=5627) and K=100 bases, the matrix size for the phase would be of 576,767,500 numpy 64-bit complex numbers. The computational power needed makes the algorithm very inaccessible for a normal PC, especially knowing that even 100 bases is a small number for this purpose (to separate the audio of a beehive may be enough but not for other types of audio like speech). It is highly illustrating how the author of the implementation that was used in this project capped the maximum number of time points to 100 to run the program.

With the limitation that the author set, the results obtained after factoring a few random pieces of audio were within the limits of the ones that were recorded for NMF (results that will be shown later in this document). Of course, trying an algorithm like this one with a low number of samples is not significant enough to compare it with the rest of the methods discussed in this paper.

So, even if the results recorded in literature about this algorithm are better than the ones with other variants of NMF, a compromise between what is needed in terms of audio separation quality and its cost in terms of computational power and time has to be met. For this reason this method was discarded for the experimentation phase of the project.

## F- CNMF-WISA

### F.0- Introduction

In his work [17], Brian King proposes two alternative algorithms for solving the problems mentioned in the previous section about complex nonnegative matrix factorization. These two methods are based on hard assumptions that we wanted to check to be good fitting in our project, as the advantages of this new variant can be very appealing if its performance is better or at least good enough.

For this purpose, we will make a revision of the first of these methods: complex nonnegative matrix factorization with intra-source additivity (or CNMF-WISA).

**F.1- Description**

The alternative method proposed is based on a big assumption: the components belonging to a single source can share the same phase information. This obviously solves in part the problem of overparameterization that was mentioned above, as now one dimension of the phase matrices is drastically reduced in size; however, the operations still require much more computational power than the other compared methods.

Some parts of the algorithm remain similar to its predecessor, although the operations are now performed differently in many cases (some spectrogram-like matrices are now grouped by source instead of having independent matrices for each base component).

To start the MM algorithm for informed source separation, W must be a set of known values, H must be initialized with random values around 1, and the matrix Φ with the phase information is initialized to the same values of the observed matrix V. After this, the values of β, $\overline{X}$, Φ and H are updated iteratively until a threshold is satisfied or a number of repetitions is completed. Here β is a 3 dimensional nonnegative matrix whose summation of all elements must be lower than 1, but it is updated to optimize the process (this β does not refer to the divergence explained earlier). $\overline{X}$ is the auxiliary function used by the algorithm in the majorization step, and Φ and H are updated in the minimization step.

About the actual code, it may be relevant to note that most of the operations between matrices were performed using Numpy's einsum, except whenever Numpy's dot could be used, as it proves to be faster for matricial products in these dimensions.

In the following section the results obtained from this method will be compared with the other algorithms studied in this project.

## G- Comparison and results

In this section 3 implementations were compared to discuss the performance of β-NMF (β=1), NMF (Euclidean distance) and CNMF-WISA. It must be noted that one of the the main initial objectives of the project was to analyze the original CNMF algorithm, although after testing it, its requirements made it very difficult to obtain results. It was later decided to test and compare only the other algorithms discussed. It is important to note that even if CNMF-WISA is a method proposed to solve the overparameterization problem inherent in CNMF, performing the same task takes up to 200 times more computational power than β-NMF and NMF.

To compare the methods, a data set was prepared with 100 audios of 1 minute duration recorded in the BeeMon project. The audios were selected to be clean, without any major sources of noise (such as strong wind, rain or human noises). The audios were sliced into 30 second pieces, half of them mixed with car noise and half of them with wind noise, properly adapting the power to make the RMS of both sources similar. The final dataset was composed of 200 original recordings and 200 mixed audios.

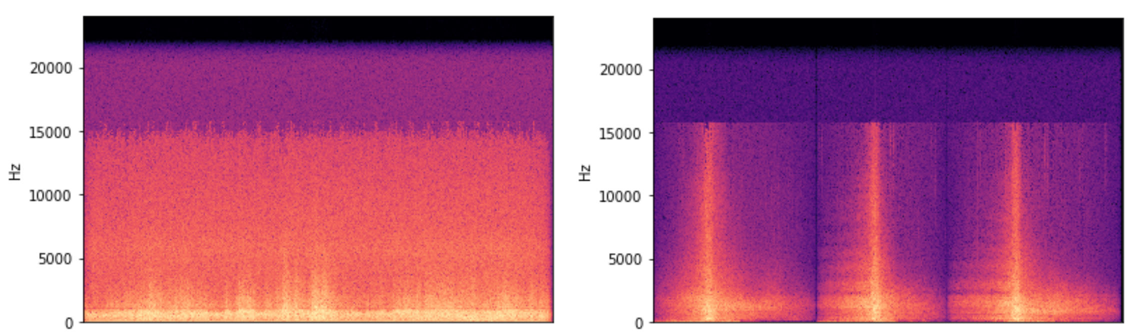
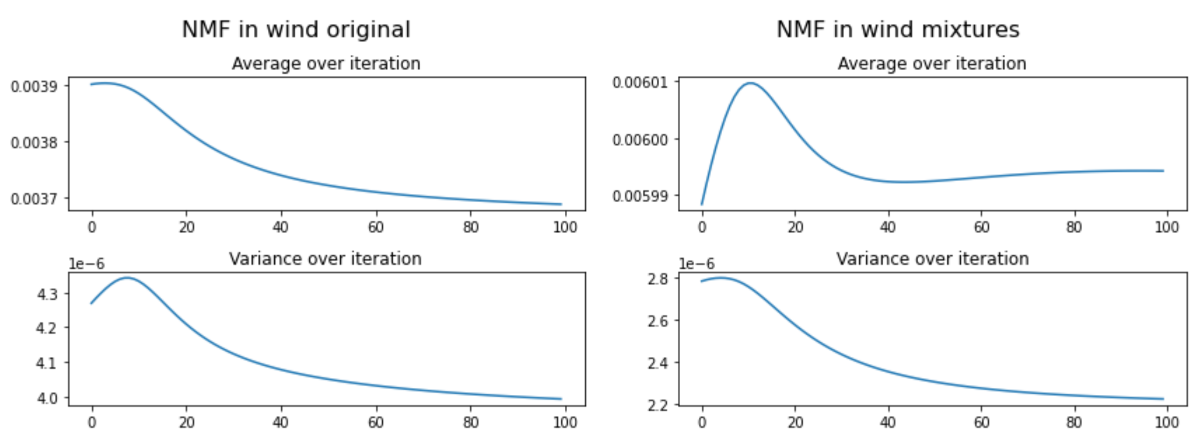
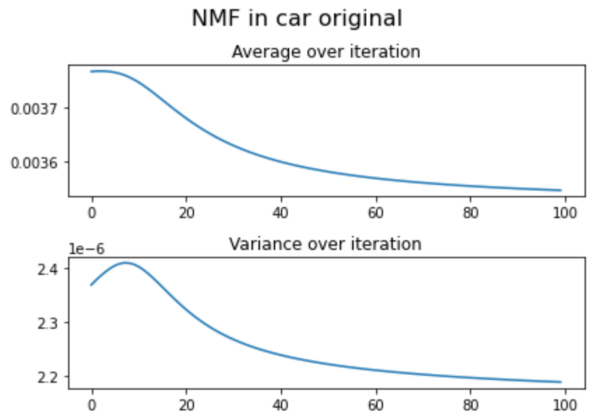


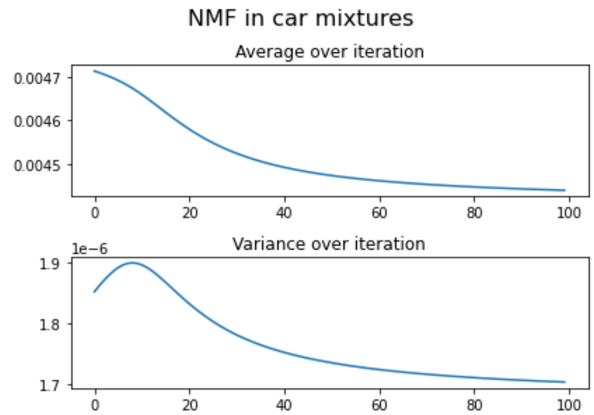

Fig 3. Wind and car added noise spectrograms with linear y-axis to better compare with future figures

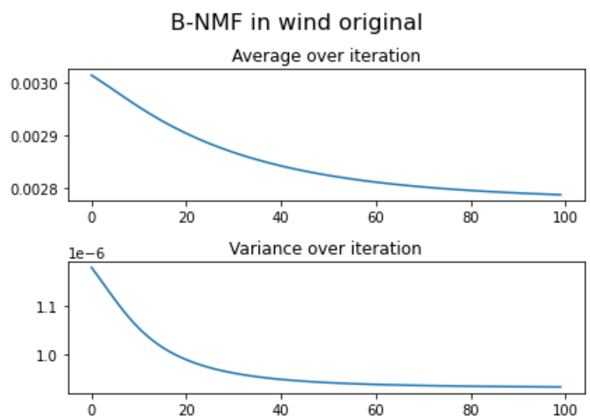
The way it is designed is to be able to compare, not only how well can these methods factorize a single matrix, but also how well can they separate a single source using appropriate bases. Here, a good reconstruction would imply low error between the reconstructed and the original spectrograms for clean audios, and a good separation would imply low error between the reconstructed spectrograms obtained from the mixture and the original audios. All of the factorizations were made updating only the temporal activations matrices, using the fixed spectral bases obtained as was explained in section C.3.

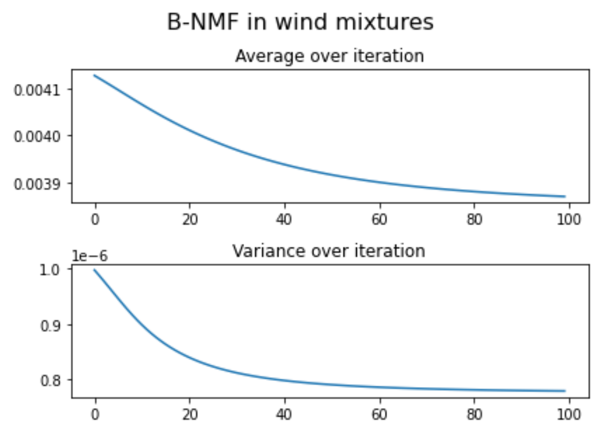
After obtaining the results, multiple statistical tests were performed to better understand the efficacy of the methods. In this first set of figures, the average absolute cost per cell for each 100 spectrograms is shown, as well as its variance. A low variance would mean that the average cost per cell for every spectrogram analyzed converges to similar values. Presumably, a good algorithm would start at a higher variance (as the activations are initialized at random values and so the first iteration is not as effective for every pair of matrices), and it would converge to a lower variance (as for similar sounds the final reconstructions may be similarly fitting after enough iterations). In this case a decreasing variance would correspond to a converging algorithm and an increasing variance would correspond to an inconsistent algorithm.

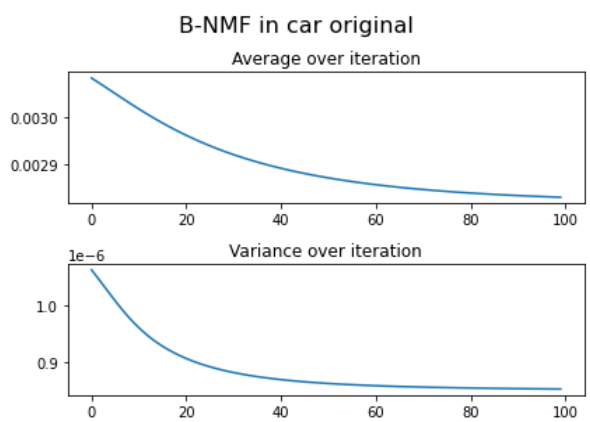


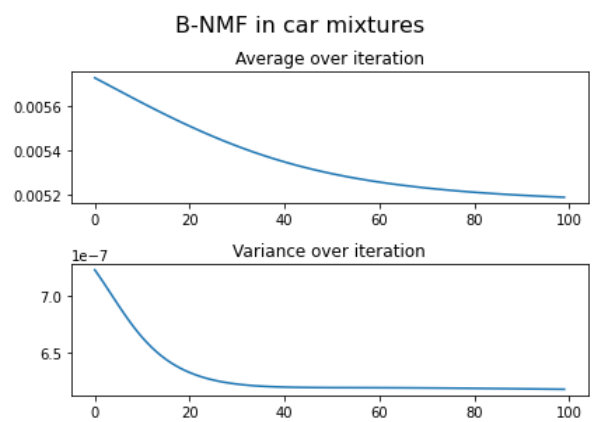

NMF in car original
Average over iteration
Variance over iteration
1e−6


NMF in car mixtures
Average over iteration
Variance over iteration
1e−6


B-NMF in wind original
Average over iteration
Variance over iteration
1e−6


B-NMF in wind mixtures
Average over iteration
Variance over iteration
1e−6


B-NMF in car original
Average over iteration
Variance over iteration
1e−6


B-NMF in car mixtures
Average over iteration
Variance over iteration
1e−7

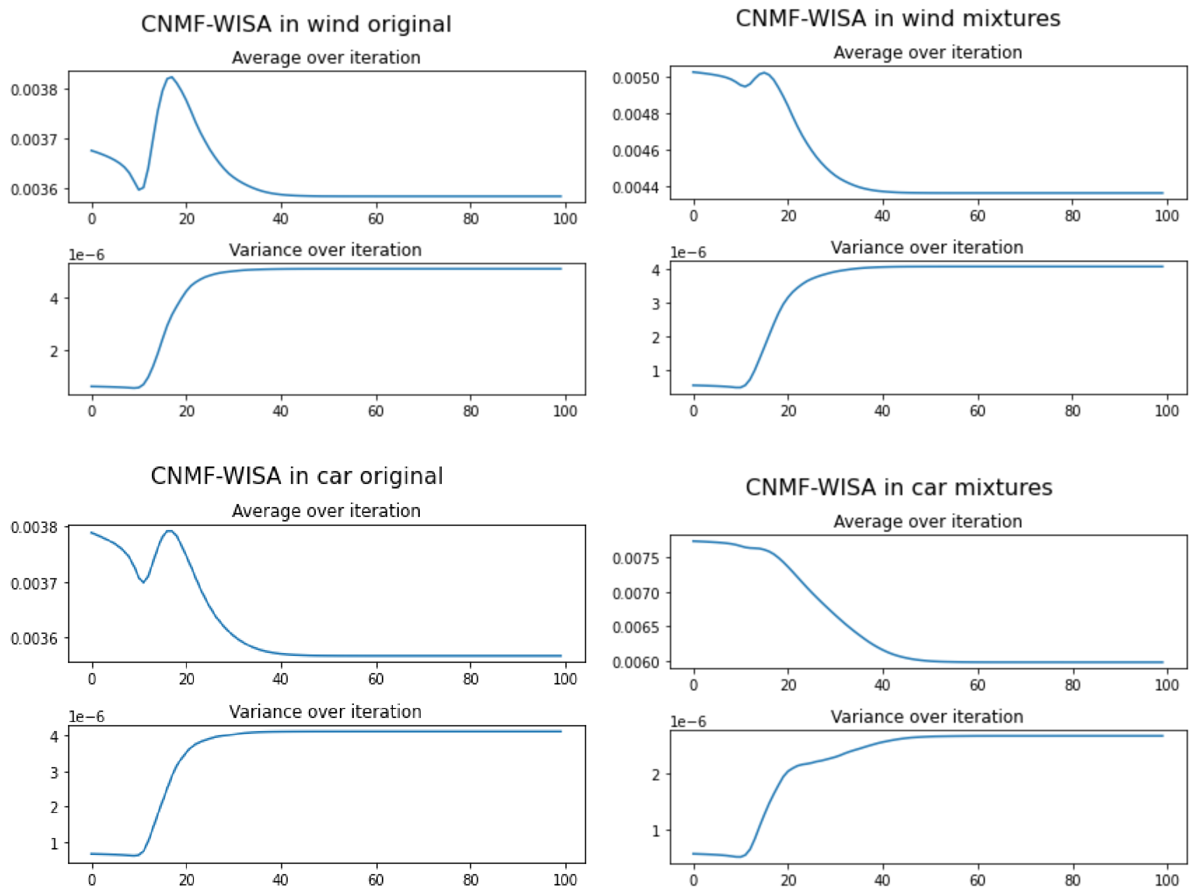



Fig 4. Average and variance of cost over iteration when factoring the original and the mixed audios with the 3 techniques discussed in this section

For the sake of comparison, the average cell value in a non mixed audio is 0.004, in the wind mixtures it is 0.007 and in the car mixtures 0.006. However, in these figures we need to focus on how the learning curves develop; removing any background noise would lead to high reconstruction error but would be considered a good separation of the targeted source.

With these experiments it is shown that this implementation of CNMF-WISA leads to non-monotonic cost functions and increasing variances. Even if the algorithm converges in around 40 iterations to lower costs, with this method the worst results are obtained; not only because of its higher costs but also because of its increasing variance, leading to inconsistency. The best results are obtained using β-NMF, with lower costs and decreasing variance. However, this only proves the method to be better at reconstructing matrices with the provided bases.

The next figures show normalized distributions of the difference between a cell in the reconstruction of an original spectrogram and the same cell in the reconstruction of the mixture of the same audio. The same analysis is performed for each of the methods proposed.

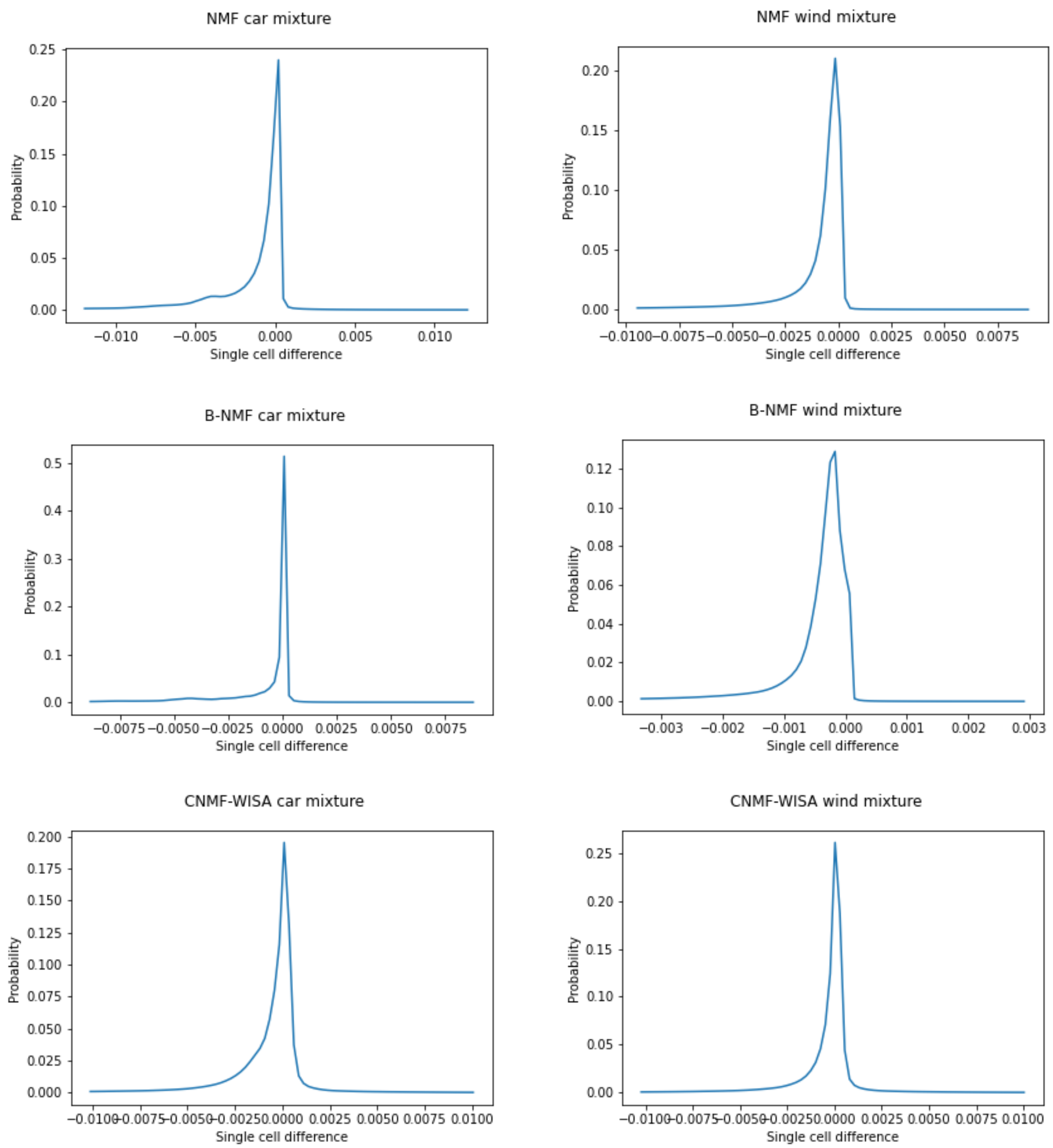



Fig 5. Distribution of the difference between the reconstruction of a cell from an original audio and its mixture.

Several properties that can be observed in these distributions:

1. They are centered around 0 and unimodal.
2. They are more heavily tailed to the negative values.
3. The smallest tails are found in β-NMF.

The first property means that, whenever the one mode is fairly narrow, the reconstruction and separation is good; there are no modes that represent frequential components being added or removed incorrectly and the error is not widely spread .

The second property can be explained with two reasons. The first being that the added noise is generating an output that overestimates the sound produced by the beehive, but does not affect its frequential shape, as will be shown in the next figures. The second reason is the following inequality referring to the addition of complex spectrograms:

$$|S1| + |S2| \geq S1 + S2 \tag{8}$$

Since the bases are absolute values obtained to describe complex spectral information, their combination over time to approximate a spectrogram tends to overestimate its power.

The reason behind the third property is related to the previous point. The β-divergence for a value of β=2 is symmetric, which means that values greater or lower than the target are equally penalized. However, as it was seen, the algorithms tend to overestimate the values in the spectrogram. For values β<2, the algorithm better penalizes the overestimation, proving to achieve more accuracy for the Kullback Leibler divergence than the Euclidean distance.

Lastly, the only complex factorization method, since it introduces phase information, it would be more likely than the others to underestimate the results. However, its intra-source additivity in single source separation makes it irrelevant, and we can still appreciate harder negative tails in the distributions calculated.

The purpose of the next set of figures is to analyze if the error found while separating a single source follows a pattern over frequency, as controlled error is much more convenient in any data processing task.  For a better understanding of the figures, we will first show the ratio of the original spectrograms divided by their mixtures, averaged over frequency.

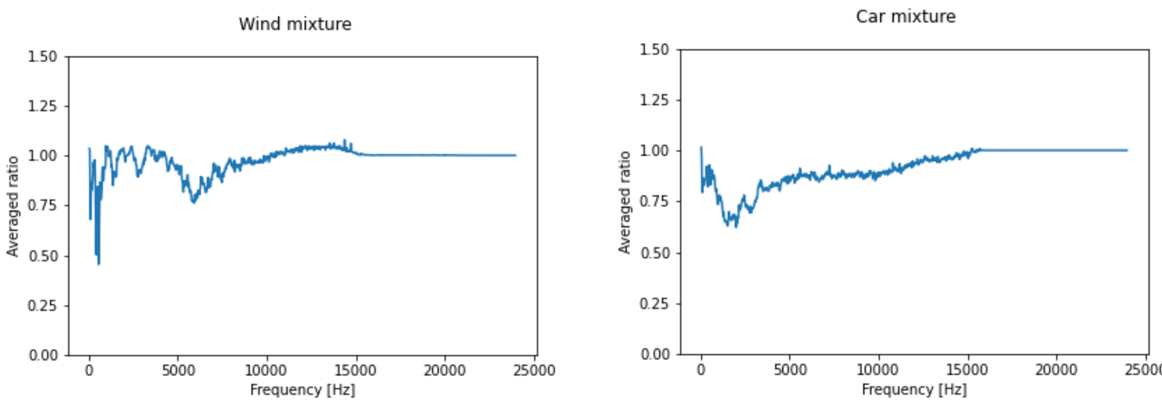
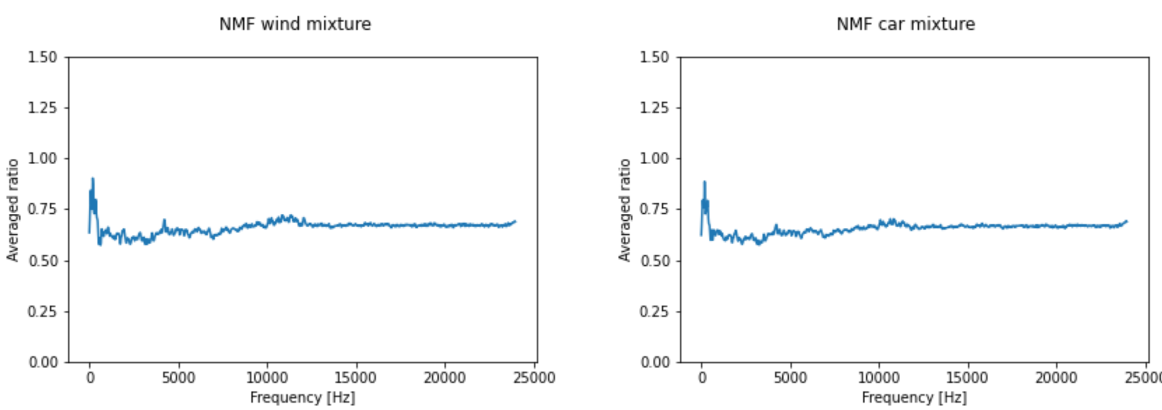



Fig 6. Averaged ratio over frequency between the original spectrograms and their mixtures.

Comparing these figures with the spectrograms shown previously, it is clear how the mixed components affect the outcome of the ratio. This last set of figures represents the averaged ratio over frequency between the original audios' reconstructions and the ones from the mixed audios.



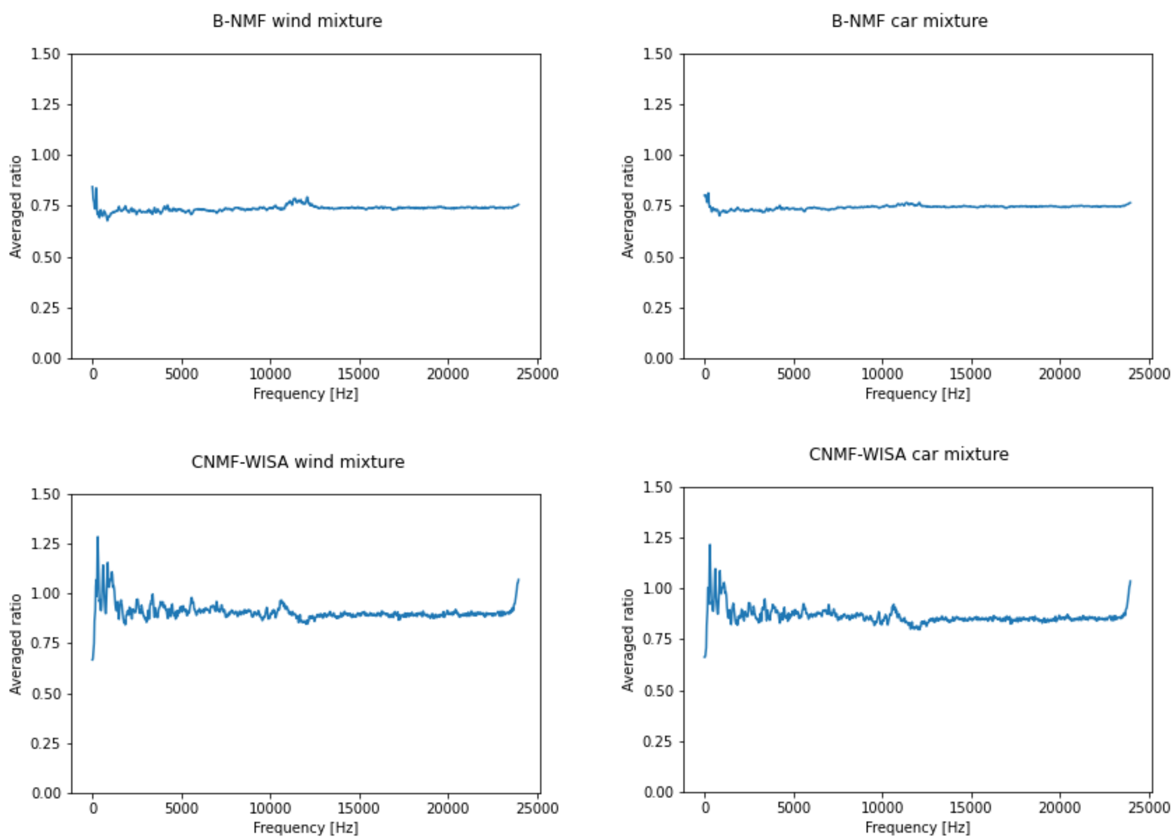



Fig 7. Averaged ratio over frequency between the reconstructed original spectrograms and their reconstructed mixtures.

It can be seen that in all cases the reconstruction of the mixtures tends to overestimate the outcome while trying to respect the frequential shape that the basis can describe. However, again the best results are proven to be delivered by the β-NMF implementation. In this set of results, there is still some residual error that belongs to the sources of noise. However, in the rest of the frequency range the mixed noise seems to disappear. Combining these figures and the previous one we can also tell that NMF seems to be able to separate the car noise but does not perform well with the wind mixtures.

Since the reconstruction of a mixed signal leads to similar frequency distributions as reconstructing an original recording, we can tell that the sources of noise were correctly eliminated, and since the bases describe well the sound produced by the beehive, the outcome is representative. The decreasing variance in the error over iteration of the experiments shown also proves the consistency of the method.

## H- Conclusion and future work

In this paper multiple methods of matrix factorization for beehive audio separation were discussed and tried. As it was seen, the best of the algorithms that were implemented was the β-NMF for $\beta=1$. The lack of symmetry in its cost function, the β-divergence, proved to be useful in factoring the absolute value of spectrograms. To reconstruct the audio recordings with the separated source, the phase of the original spectrogram could be copied (which would maintain the same level of error) or other methods to calculate a phase could be used (which would increase the final reconstruction cost but could be better for the separated audio).

It was also seen in this work that the overparameterization of the CNMF makes it difficult to experiment with in a regular PC, especially with the dimensions required in real audio spectrograms. As an alternative, CNMF with intra-source additivity was proposed and implemented, seeming an attractive alternative that included phase information while partly solving the problems of its predecessor. However, the results provided by its implementation were not as satisfactory as the ones from the methods tested during this time. A reason for this could be that the assumption that every base used to characterize a source of audio can share the same phase information does not apply to this precise case.

Some other alternatives that include phase information in the factorization of spectrograms could be studied, like the ones using phase constraints (phase evolution or phase unwrapping). It would be also very beneficial to implement these methods in TensorFlow, to improve the operation with the volume of matrices that is needed.

It would also be helpful to the development of the project to characterize and categorize anomalies in the frequency distribution of the sound produced by the beehive; it could be used to form bases whose presence in the reconstruction of an audio would help detect abnormal conditions.

## I- References

[1] Computer Science Department, Appalachian State University. Beemon. Retrieved 2022, from https://cs.appstate.edu/beemon/

[2] HUNTER, David R.; LANGE, Kenneth. A tutorial on MM algorithms. The American Statistician, 2004, vol. 58, no 1, p. 30-37.

[3] WANG, Yu-Xiong; ZHANG, Yu-Jin. Nonnegative matrix factorization: A comprehensive review. IEEE Transactions on knowledge and data engineering, 2012, vol. 25, no 6, p. 1336-1353.

[4] LIU, Weixiang; ZHENG, Nanning; YOU, Qubo. Nonnegative matrix factorization and its applications in pattern recognition. Chinese Science Bulletin, 2006, vol. 51, no 1, p. 7-18.

[5] CHU, Moody; PLEMMONS, Robert. Nonnegative matrix factorization and applications. Bulletin of the International Linear Algebra Society, 2005, vol. 34, no 2-7, p. 26.

[6] SEUNG, D.; LEE, L. Algorithms for non-negative matrix factorization. Advances in neural information processing systems, 2001, vol. 13, p. 556-562.

[7] YOSHII, Kazuyoshi, et al. Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction. En ISMIR. 2013. p. 369-374.

[8] Beginaid. (n.d.). ComplexNMF. GitHub. Retrieved 2022, from https://github.com/beginaid/ComplexNMF

[9] BASU, Ayanendranath, et al. Robust and efficient estimation by minimising a density power divergence. Biometrika, 1998, vol. 85, no 3, p. 549-559.

[10] OLAYA, Jbari; OTMAN, Chakkor. Beta-divergence for Nonnegative Matrix Factorization. En 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA). IEEE, 2021. p. 15-22.

[11] FÉVOTTE, Cédric; IDIER, Jérôme. Algorithms for nonnegative matrix factorization with the β-divergence. Neural computation, 2011, vol. 23, no 9, p. 2421-2456.

[12] MAGRON, Paul; BADEAU, Roland; DAVID, Bertrand. Phase recovery in NMF for audio source separation: an insightful benchmark. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015. p. 81-85.

[13] KAMEOKA, Hirokazu, et al. Complex NMF: A new sparse representation for acoustic signals. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009. p. 3437-3440.

[14] KAMEOKA, Hirokazu; KAGAMI, Hideaki; YUKAWA, Masahiro. Complex NMF with the generalized Kullback-Leibler divergence. En 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017. p. 56-60.

[15] MAGRON, Paul; BADEAU, Roland; DAVID, Bertrand. Complex NMF under phase constraints based on signal modeling: application to audio source separation. En 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016. p. 46-50.

[16] BRONSON, James; DEPALLE, Philippe. Phase constrained complex NMF: Separating overlapping partials in mixtures of harmonic musical sources. En 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014. p. 7475-7479.

[17] KING, Brian John. New methods of complex matrix factorization for single-channel source separation and analysis. University of Washington, 2012.