

ALMA MATER STUDIORUM · UNIVERSITY OF BOLOGNA

---

Statistical and Mathematical Methods for Artificial Intelligence

**Blind Signal Separation (BSS) Problem solved with NMF.**

Lorenzo Cellini ([lorenzo.cellini3@studio.unibo.it](mailto:lorenzo.cellini3@studio.unibo.it))

February 5, 2022

# Contents

1	Summary	2
2	Background theory	3
3	Exercise	4
4	Conclusion	7

# 1 Summary

Blind signal separation (BSS) is the task of separating a set of source signals from a set of mixed signals, without the aid of information (or with very little information) about the source signals or the mixing process. An example of BSS problem is the separation of musical instrument tracks when playing together or the most common "cocktail party" example, where a group of people are talking (during a cocktail party) and one wants to follow one of the discussions. This is in general a very easy task for human brain, although not easy for machines.

In this work, BSS problem has been addressed through Non-negative Matrix Factorization (**NMF**) technique: a group of algorithms employed in multivariate analysis and linear algebra that shares common traits with clustering techniques.

The given dataset consists of 20 recording tracks each of 1000 time step length. These are recorded by 20 different microphones and contain a mixture of the tracks played by 4 different instruments. Applying NMF to the given dataset it is possible to separate the musical instruments tracks, and so it is possible to detect how many (and maybe also what kind of) instruments are playing.

Experiments with a different requested number of instruments have been carried out in order to simulate and compare the algorithm behaviour, when the true number of sources is not known.

## 2 Background theory

Non-negative Matrix Factorization (or Approximation) consists of factorizing a matrix  $\mathbf{X}$  into two matrices  $\mathbf{W}$  and  $\mathbf{H}$ , where all the three matrices have no negative elements. The non negativity makes these matrices easier to inspect and interpret and sometimes is inherent to the real problem itself. NMF can be seen as computing the columns of  $\mathbf{X}$  as linear combination of the columns of  $\mathbf{W}$  with coefficient of the combination taken from  $\mathbf{H}$ :

$$\vec{x}_i = \mathbf{W}\vec{h}_i \quad (2.1)$$

The dimensions of the factor matrices  $\mathbf{W}$  and  $\mathbf{H}$  may be significantly lower than those of the original matrix  $\mathbf{X}$  and this is a very nice property because it makes NMF a powerful technique to be applied to dimensionality reduction and clustering problems.

There are different types of non-negative matrix factorizations: the different types arise from using different cost functions for measuring the divergence between  $\mathbf{X}$  and  $\mathbf{WH}$  and possibly by regularization of the  $\mathbf{W}$  and/or  $\mathbf{H}$  matrices. In this exercise, the given NMF function implements approximation through Non-negative Least Square (NNLS) algorithm, a type of constrained least squares problem where the coefficients are not allowed to become negative.

The factorization problem in the squared error version of NMF may be stated as: given a matrix  $\mathbf{X}$  find non-negative matrices  $\mathbf{W}$  and  $\mathbf{H}$  that minimize the function

$$f(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|_F \quad (2.2)$$

subject to  $\mathbf{W}, \mathbf{H} \geq 0$ .

After the factorization, the columns of  $\mathbf{W}$  are the new basis for the column of  $\mathbf{X}$ , while columns of  $\mathbf{H}$  can be interpreted as the coordinates of a data point in the basis  $\mathbf{W}$ .

Making a parallel with respect to clustering techniques, the column of  $\mathbf{W}$  represents the cluster centroids and so, the number of columns of  $\mathbf{W}$  is the number of clusters found.

### 3 Exercise

In this exercise I have implemented three different experiment: in the first one I have experimented with different values of  $m$ , the number of sources to be found (also clusters); in the second one, after having fixed  $m=4$ , I have tried different values for  $\tau$ , that is the relative difference between two consecutive step of the algorithm, used in order to check for convergence; in the third one I have used NMF function from sci-kit learn package and compared results of the two algorithms.

Results for the different values of  $m$  are shown in 3.1, 3.2 and 3.3.

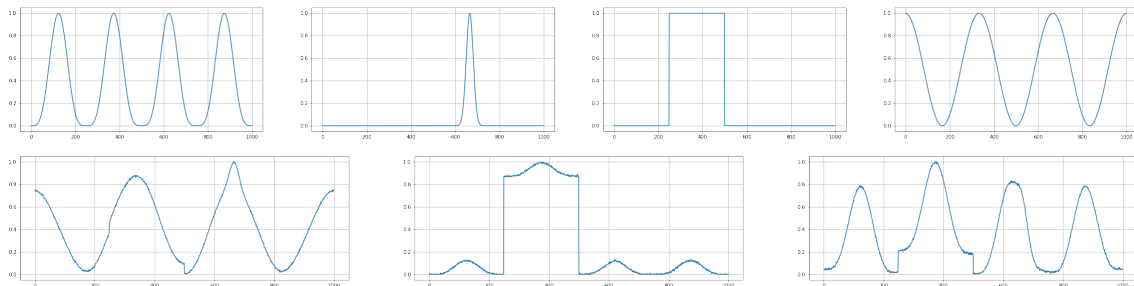


Figure 3.1: Source detected with  $m=3$ .

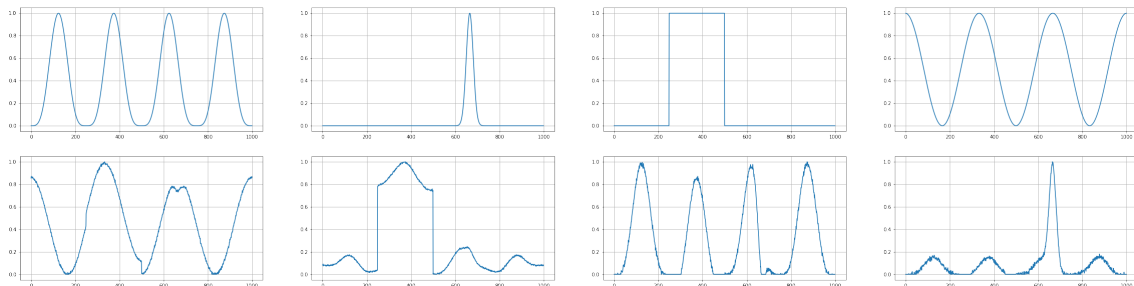


Figure 3.2: Source detected with  $m=4$ .

As it can be seen, the best result is when  $m=4$ , that corresponds to the true number of sources. The ordering of the solution does not matter here because it is always possible to re-arrange matrix column without changing the matrix itself. In particular, when using a smaller number of sources, it can be observed how the true fourth one is mixed with the detected three, e.g. by looking at the third peak of the first plot. When asking for 5 sources instead, the algorithm creates a fifth noisy wave and also the other four become more noisy because some points are not assigned anymore to them but they are assigned to the new one.

Now, fixing the number of sources to 4, I tried the same algorithm with the three different values for  $\tau$  [1e-2, 1e-3, 1e-4] in order to check how the number of iteration and the approximation error

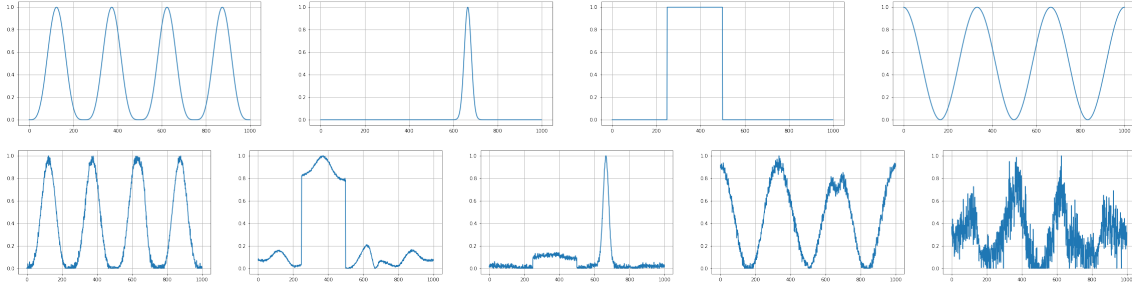


Figure 3.3: Source detected with  $m=5$ .

change. In 3.4 it can be seen how the number of iteration increases when  $\tau$  becomes smaller: this is quite obvious because the algorithm needs more iteration in order to reach points where the changes between two iteration are smaller than  $\tau$ . What is interesting is that the minimum of the reconstruction error, namely the Frobenius norm  $\|X - WH\|_F$ , is reached after only 5 iterations. Although there is no visual difference between the solution after 5 iteration and after 300, from a numerical point of view one should keep the solution that minimize the error and so the one corresponding to the fifth iteration. In this case, the smallest reconstruction error is around 33.

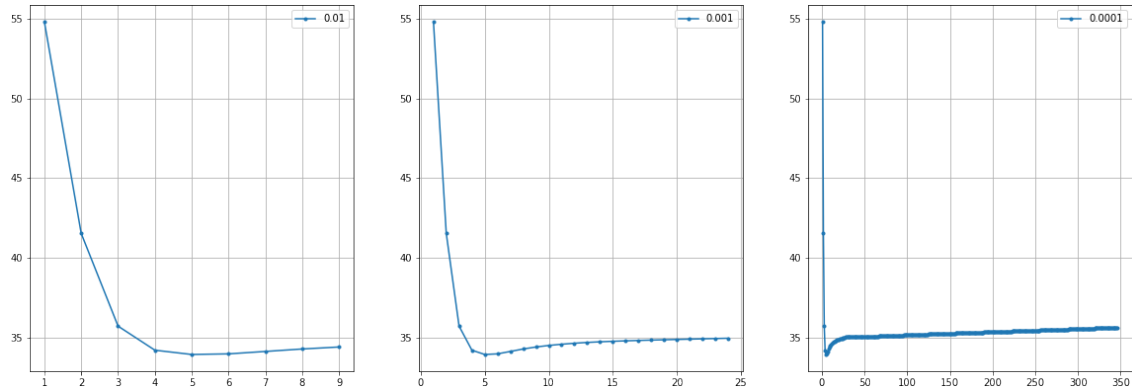


Figure 3.4: NMF reconstruction error for different values of  $\tau$ .

Finally, I have run the same experiment using *sci-kit learn* function NMF in order to compare performances of the two algorithm. The differences in the two NMF functions are in the implemented solver: *sci-kit* function has the two solvers "cd" for "coordinate descent" and "mu" for "multiplicative update", while the provided custom function relies on non-negative least square algorithm. All the three are iterative optimization algorithms.

In 3.5 it can be seen that *sci-kit* implementation outperform the custom one, and that coordinate descent method is more effective than multiplicative update: indeed the first reaches a better minimum of around 2.5 within 264 iteration, with respect to the latter that reaches a minimum of 2.9 after 269 iterations. While *sci-kit* implementation shows a smaller reconstruction error, the visual inspection of detected sources is not as good as for the custom NMF.

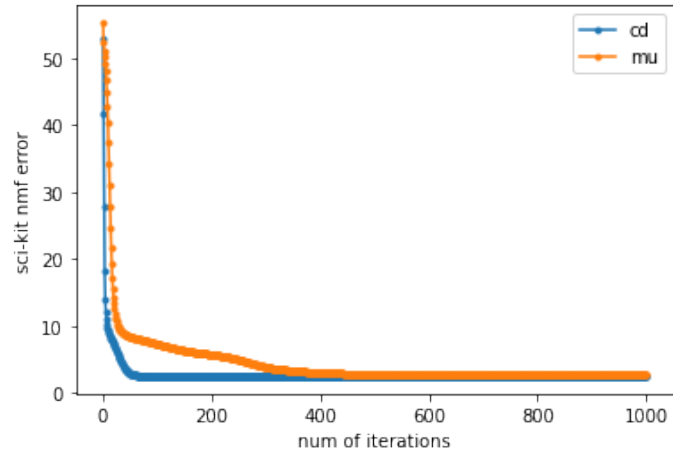


Figure 3.5: Sci-kit NMF reconstruction error for different solvers.

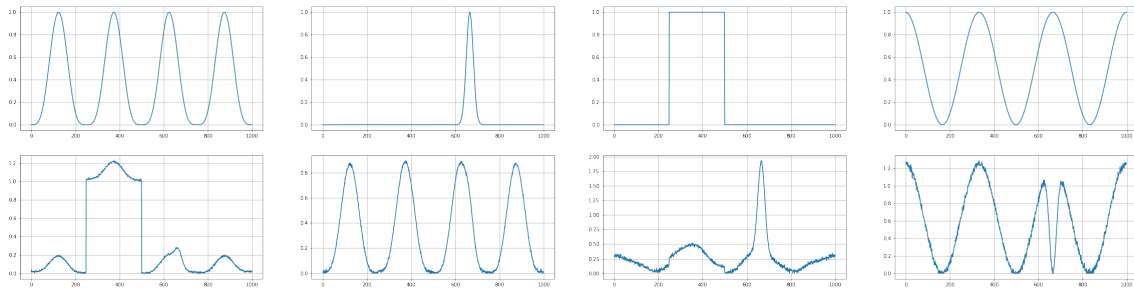


Figure 3.6: Sci-kit NMF source detected for solver "cd".

## 4 Conclusion

NMF has proven to be very powerful in clustering and dimensionality reduction problems like image or sound signal separation, topic extraction in text mining and so on. As shown in the different experiment, there are a lot of parameters to take into account when running NMF and so, as usual in machine learning, the best approach would be to perform a search grid on the space of parameters in order to find the best configuration for each problem.

Being clustering typically an unsupervised techniques, the number of sources to be detected is not known in principle and so the procedure should be to try for different values of input  $m$  and asses results both visually and trough the definition of a fitness function (like in K-means clustering).