# NMF applications to pitch estimation

Project Report MATH-412
Mariia Soroka, Tâm Johan Nguyên, Vireak Prou
*January 2022*

***Abstract*—Non-negative matrix factorisation (NMF) alogrithms allows to factorize matrices while enforcing that the coefficients in the decomposition are non-negative. In the case of spectogram factorisation this allows for meaningful separation into sound components. Here we consider multiplicative gradient and expectation-maximisation (EM) algorithms for factorisation based on $\beta$-divergence and study how they perform under different choices of initialisation and $\beta$. In particular we show applications of NMF to pitch estimation and denoising of audio recording and study how those different methods lead to identification of different sound features.**

## I. Introduction

### A. Non-negative matrix factorization

Given a matrix $V \in \mathbb{R}^{F\times N}$ such that $V_{fn} \geq 0$ for $1 \leq f \leq F$ and $1 \leq n \leq N$, the non-negative matrix factorization (NMF) of $V$ is a low-rank approximation where we are looking for $W \in \mathbb{R}^{F\times K}$, $H \in \mathbb{R}^{K\times N}$ with non-negatives entries for $W$ and $H$ such that $V \approx WH$. Usually, we choose $K$ such that $FK + KN < FN$ in order to reduce the dimension of $V$. Unlike other low-rank approximation like principal component analysis (PCA) or singular value decomposition (SVD), the unique feature of NMF is that it enforces non-negative coefficients which can be more appropriate for some applications: in our case spectrogram factorisation.

NMF of $V$ can be expressed as the following minimisation problem:

$$\min_{W,H\geq 0} \; D(V|WH)$$

for some cost function $D(V|WH) := \sum_{f=1}^{F}\sum_{n=1}^{N} d(V_{fn}|(WH)_{fn})$. For this project we will study this problem for $d(\cdot \mid \cdot)$ in a family of divergences called $\beta$-divergences. Several algorithms were created using these cost functions, in this report we will cover two of them taken from (1).

### B. $\beta$-divergences

For $\beta \in \mathbb{R}$ we define the $\beta$-divergences $d_\beta : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ as:

$$d_\beta(x|y) := \begin{cases} \frac{x}{y} - \log\frac{x}{y} - 1 \text{ if } \beta = 0 \\ x(\log x - \log y) + (y - x) \text{ if } \beta = 1 \\ \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}) \text{ if } \beta \in \mathbb{R}\backslash\{0,1\} \end{cases}$$

Some remarkable cases occurs when $\beta = 2$ in which we get the Euclidian (EUC) distance, $\beta = 1$ the Kullback-Leiber (KL) divergence, and $\beta = 0$ the Itakura-Saito (IS) divergence.

For all $\beta$ we note that $d_\beta(x|y) = 0$ whenever $x = y$ and also that its derivative relative to $y$ is $\nabla_y d_\beta(x|y) = y^{\beta-2}(y-x)$. This shows that $d_\beta(x|y)$ has a unique minimum at $x = y$ and increases with $|x - y|$. This justifies the choice of $\beta$-divergences as cost function.

### C. Application to pitch estimation and denoising

We now explain how we can use NMF to decompose sound. We represent the sound of interest as a power spectogram $V$. This matrix captures the frequencies of the sound and their power over time, by its nature $V$ is non-negative. We can apply a NMF to $V$, the matrices obtained $W$ and $H$ both have a musical interpretation. The columns $w_k \in \mathbb{R}^F$ of $W$ represent the frequency spectra of the $k$-th sound component (a pitch typically) discretized into $F$ frequency bins and the rows $h_k \in \mathbb{R}^N$ of $H$ corresponds to the amplitude over time (in $N$ time bins) of this $k$-th sound component.

This decomposition reveals to be useful for pitch estimation, we can recover an estimation of the pitches present in some spectrogram by taking $w_k \cdot J$ for $J$ some frequency combs, the $J$ such that this scalar product is maximised corresponds to the pitch/frequency identified. Here $J$ the frequency comb for some frequency $f_j$ is simply a cosinus with frequency $f_j$ correspondings to the pure harmonic. We have used MIDI notation for pitches, but the following formula can be used to convert MIDI pitch $p$ to the sound frequency $f$ in Hz: $f = 440\ 2^{\frac{p-69}{12}}$.

Another application of NMF is denoising, indeed after factorisation some sound components may correspond to noise (either very low or high frequency) and thus can be removed from $W$ and $H$ (truncating them) and recomputing $WH$ gives us a denoised spectrogram of the original audio.

As expected, the divergence used for the cost function heavily affect the result we get. If we choose to minimize the euclidean distance, we get a "smoother" reconstruction because the low power components tends to be ignored. On

the contrary, the IS divergence is scale invariant that is for $\gamma \in \mathbb{R}$, $d_0(\gamma x|\gamma y) = d_0(x|y)$. Because of this property an IS-NMF gives the same importance to high and low power components.

In this report we are going to use two different algorithms to solve NMF, one using multiplicative gradients updates and the other using expectation maximisation. Then we show some results for pitch estimations and denoising when applied to some audio recording and synthetic data.

### D. Notations

We use the notations of (1), so if $A, B \in \mathbb{R}^{F\times N}$ then $A.B$ is the element-wise product, $A^{\cdot\lambda}$ is the element-wise power $\lambda$, $|A|$ is the element-wise absolute value and $\frac{A}{B} = A.B^{\cdot -1}$

## II. Methods

All the code used for this report can be found in https://github.com/mariasoroka/StatisticalMachineLearning. In particular the algorithms as well as some useful functions to convert audio to spectrogram and spectrogram to audio can be found in the file NMF.py.

### A. Multiplicative gradient

The first algorithm for NMF that we present is based on a multiplicative update gradient method. The algortihm is shown to be non increasing for $1 \leq \beta \leq 2$, and seems in practice to be non increasing for others $\beta$ but no proof is yet provided.
The proof of convergence of the algorithm is based on the non increasing property of auxiliary functions.

**Definition: auxiliary function.**
$g(y, \tilde{y})$ is an auxiliary function for the cost function $D(y)$ if for every $y$ and $\tilde{y}$:

$$g(y, y) = D(y) \tag{1}$$
$$g(y, \tilde{y}) \geq D(y) \tag{2}$$

**Lemma 1.**
If $g(y, \tilde{y})$ for $D(y)$ then the update

$$y_{new} = \arg\min_y g(y, y_{old})$$

is non increasing for D(y)

The algorithm updates $W$ and $H$ by minimizing their respective auxiliary functions leading to multiplicative gradient updates. This algorithm converge to a local minimum for $\beta \in [1, 2]$.
The complete proof is reported in Appendix A.

### B. Expectation maximisation

Both the Theorem and SAGE algorithm described in this section comes from (1). The proof and derivation in the Appendices are based on their work.

The NMF of a spectogram under IS-divergence can actually be framed as a likelihood maximisation problem. Indeed if we consider a signal $X \in \mathbb{R}^{F\times N}$ ($F$ the frequencies, and $N$ the time frame) such that $X = \sum_{k=1}^{K} C_k$ where $(C_k)_{:n} \sim \mathcal{N}(0, h_{kn} \cdot diag(w_{:k}))$ represents sound component $k$ at time frame $n$ with activation coefficient $h_{kn}$ and frequency spectra $w_{:k}$ (note that the coefficients of those parameters are non-negative). Then if we wished to use maximum likelihood to derive estimates for $W$ and $H$ we are left with the problem $max_{W,H} p(X \mid W, H)$ with $p$ the likelihood.

**Theorem 1.**
*In the previous problem, $max_{W,H} p(X \mid W, H)$ is equivalent to $min_{W,H} d_{IS}(|X|^{\cdot 2} \mid WH)$ and hence is equivalent to NMF of $|X|^{\cdot 2} = V \approx WH$ under IS-divergence.*
**proof:** refer to Appendix B.

From this statistical interpretation of IS-NMF we can derive an expectation-maximation based algorithm, namely Algorithm 2 : SAGE algorithm for IS-NMF. For the complete derivation refer to Appendix C. One very important point to note is that the updates in Algorithm 2 forces the coefficients of $W$ and $H$ to be strictly positive, hence we can only converge toward a factorisation with strictly positive coefficients.

### C. Algorithms

For convenience we refer to Algorithm 1 as EUC, KL, IS-MU when $\beta = 2, 1, 0$ respectively and to Algorithm 2 as IS-EM.

**Algorithm 1** Multiplicative Gradient for $\beta$-divergence

**Input:** non-negative matrix $V$
**Output:** non-negative matrices $W, H$ such that $V \approx WH$

```
initialize W, H with non-negative values
for i = 1 : n_iter do
    H ← H.[Wᵀ((WH)^{·β−2}.V)] / [Wᵀ(WH)^{·β−1}]        ▷ Update
    W ← W.[((WH)^{·β−2}.V)Hᵀ] / [(WH)^{·β−1}Hᵀ]

    for k = 1 : K do                                    ▷ Normalisation
        H_{k:} ← H_{k:} · ||W_{:k}||_2^2
        W_{:k} ← W_{:k} / ||W_{:k}||_2^2
    end for
end for
```

**Algorithm 2** SAGE algorithm for IS-NMF

**Input:** non-negative matrix $V$

**Output:** non-negative matrices $W, H$ such that $V \approx WH$

initialize $W, H$ with non-negative values

**for** $i = 1 : n_{iter}$ **do**

**for** $k = 1 : K$ **do**

$G_k = \frac{W_{:k} \cdot H_{k:}}{WH}$ ▷ Wiener Gain

$V_k = G_k^2 . V + (1 - G_k).(W_{:k} \cdot H_{k:})$

$H_{k:} \leftarrow \frac{1}{F}(W_{:k}^{-1})^T V_k$ ▷ Update

$W_{:k} \leftarrow \frac{1}{N} V_k (H_{k:}^{-1})^T$

$H_{k:} \leftarrow H_{k:} \cdot ||W_{:k}||_2^2$ ▷ Normalisation

$W_{:k} \leftarrow W_{:k} / ||W_{:k}||_2^2$

**end for**

**end for**

## III. Results

### A. Experiments on convergence

We investigated the convergence of the algorithms for the factorisation of some fixed randomly generated $10 \times 25$ matrix with (absolute value of) normal coefficients, $K = 5$ and 5000 iterations. We were interested in particular in how fast the convergence was, if it lead to different local minima, how different random initialisation influence convergence, and finally how the different methods compare to each other. The results can be found in Fig.6.

### B. Factorisation of synthetic data

To test our implementation we used synthetic matrix $V_s = W_s H_s \in \mathbb{R}^{1025 \times 500}$ mimicking a denoised factorisation we would expect from the chord progression in the next section with $K = 4$ pitch components. Columns of $W_s$ (representing the frequency spectra of the $k$-th component) and rows of $H_s$ (representing the activation and amplitude in time of the $k$-th component) are presented on Fig.1. Convergence in cost of the various algorithms can be found in Fig.2), the resulting factorisation for KL can be found in Fig.4. We also tried a second set of synthetic data where we added a baseline to all the coefficients in $W_s$ and $H_s$ discussed above in order to prevent them from being exactly 0 as IS-EM is expected to underperform when some coefficients are 0 due to the updates in Algorithm 2.

### C. A simple chords progression

We recorded a standard chords progression on piano for a total of 12 seconds in a noisy environment. The power spectrogram was computed from the audio recording resulting in 523 time frames and 1025 frequency bins. The score of the progression and corresponding power spectrogram are shown in Fig.3.

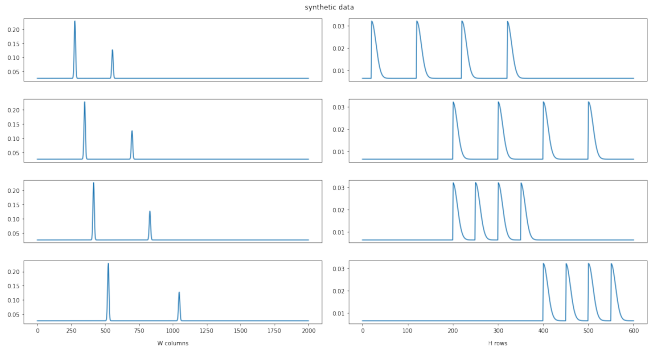


Figure 1. Synthetic audio data for $W$ and $H$ with non-negative coefficients to test factorisation of $WH$, $F = 1025, N = 500, K = 4$. From left to right : $W_{:k}$, $H_{k:}$, with $k = 0, ..., 3$ (from top to bottom).

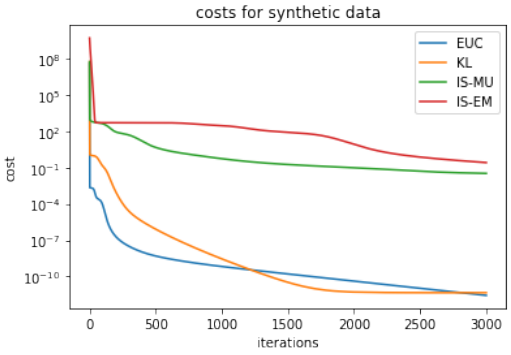


Figure 2. Cost at each iteration for different factorization algorithms with synthetically generated input matrix $V_s = W_s H_s$, $K = 4$ and 3000 iterations.

The obtained matrix was then factorized using our different algorithms with values $K$ from 5 to 8. For each resulting factorization we applied the pitch (or frequency) detection procedure detailed in the introduction.

## IV. Discussion

### A. Analysis of convergence

Regarding the convergence on some small matrix in Fig.6, EUC and KL both outperforms IS-MU and IS-EM on average by almost an order of magnitude in term of cost (almost $10^1$ for EUC and KL and around $10^2$ for IS-MU and IS-EM). The convergence in term of costs is also much more concentrated in EUC and KL, while in IS-MU and IS-EM we observe that the costs vary greatly between runs around order $10^2$. Regarding this disparity in convergence, while different initialisation does not affect EUC and KL, it somewhat helps reducing the range to which they converge when choosing larger amplitude and baseline for $H$ in both IS-MU and IS-EM. All algorithm exhibits a sharp drop in cost at the beginning, and then a more gentle decreasing curve which stabilises early on mostly before the 1000-th iteration. Although we do observe for some of the runs and

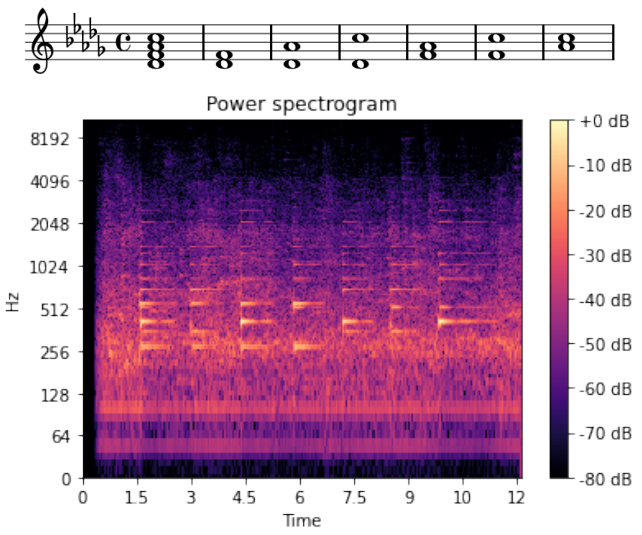


Figure 3. Piano chords progression used in the experiments and corresponding power spectrogram. The four notes read $D_4^\flat$ (pitch 61), $F_4$ (pitch 65), $A_4^\flat$ (pitch 68) and $C_5$ (pitch 72).

algorithm a later drop in cost after the cost seemed to have stabilised (IS-EM and initialisation with large amplitude and baseline).

Also note that IS-MU and IS-EM have cost increase momentarily in certain runs, contrary to what the theory tells us as the iterations should decrease the cost. This is probably due to the implementation, as our IS-divergence often suffered from overflow errors and we had to make slight adjustments to the updates to prevent them, which could have resulted

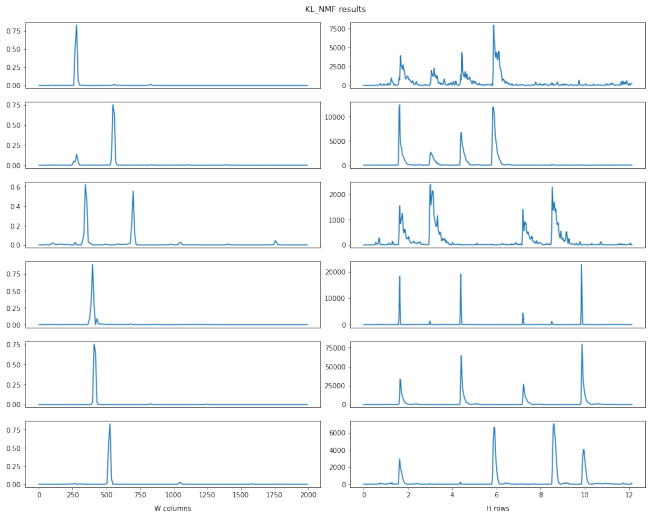


Figure 4. KL with $K = 8$. Pitch estimates in MIDI notation from top to bottom: [61, 61, 65, 67.2, 68, 72]. Corresponding frequencies in Hz: [277.2, 277.2, 349.2, 396.6, 415.3, 523.3]. Two noisy components are not plotted.

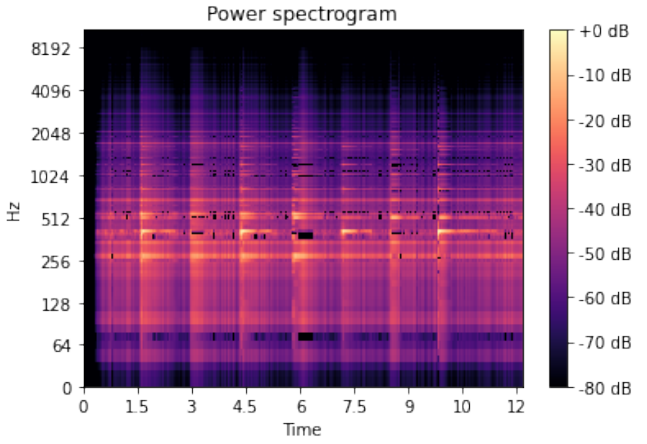


Figure 5. Spectogram for EUC factorisation with $K = 7$, original audio spectrogram in Fig.3.

in those momentary increase in costs

Overall this shows that in both case we observe some convergence to a local minimum, but that those minima are far more apart in term of costs in the IS-divergence loss leading to several runs which performed sub-optimally.

### B. *Performance on synthetic data*

Regarding the cost convergence, it seems like KL stabilised at 3000 iterations, while EUC seems to be able to continue its downward trend if we had increase the number of iterations. For IS-MU and IS-EM their costs also somewhat stabilises, but several order of magnitude above those of EUC and KL, and perhaps more iterations could have help reduced their costs further but it would not be expected to reach the magnitude of KL and EUC. Overall both KL and EUC obtain very good reconstruction at costs of order $10^{-10}$ while IS-MU and IS-EM stabilises at $10^2$ and thus comparatively provide far worse reconstruction. Based on the findings in the previous section, perhaps other run of IS-MU and IS-EM could get better results depending on which local minima they converge to, but on such a large matrix factorisation computation cost is prohibitively high and thus we were not able to explore it further.

For non-zero variant of synthetic data all the decompositions had similar behavior: all the components are correctly recovered, different components can be slightly mixed, but the amplitude of spurious modes does not exceed one fifth of the main mode amplitude. For all of the algorithms except IS-MU the amplitude of the same pitch from ”stroke” to ”stroke” lies within $15\%$ of the maximum amplitude. But for IS-MU for one of the pitches the amplitudes of successive strokes can differ by a factor of two. Otherwise we did not observe any significant difference in behavior for IS cost.

Including zeros did not affect the convergence or results' behavior of any multiplicative algorithms, but IS-EM turns

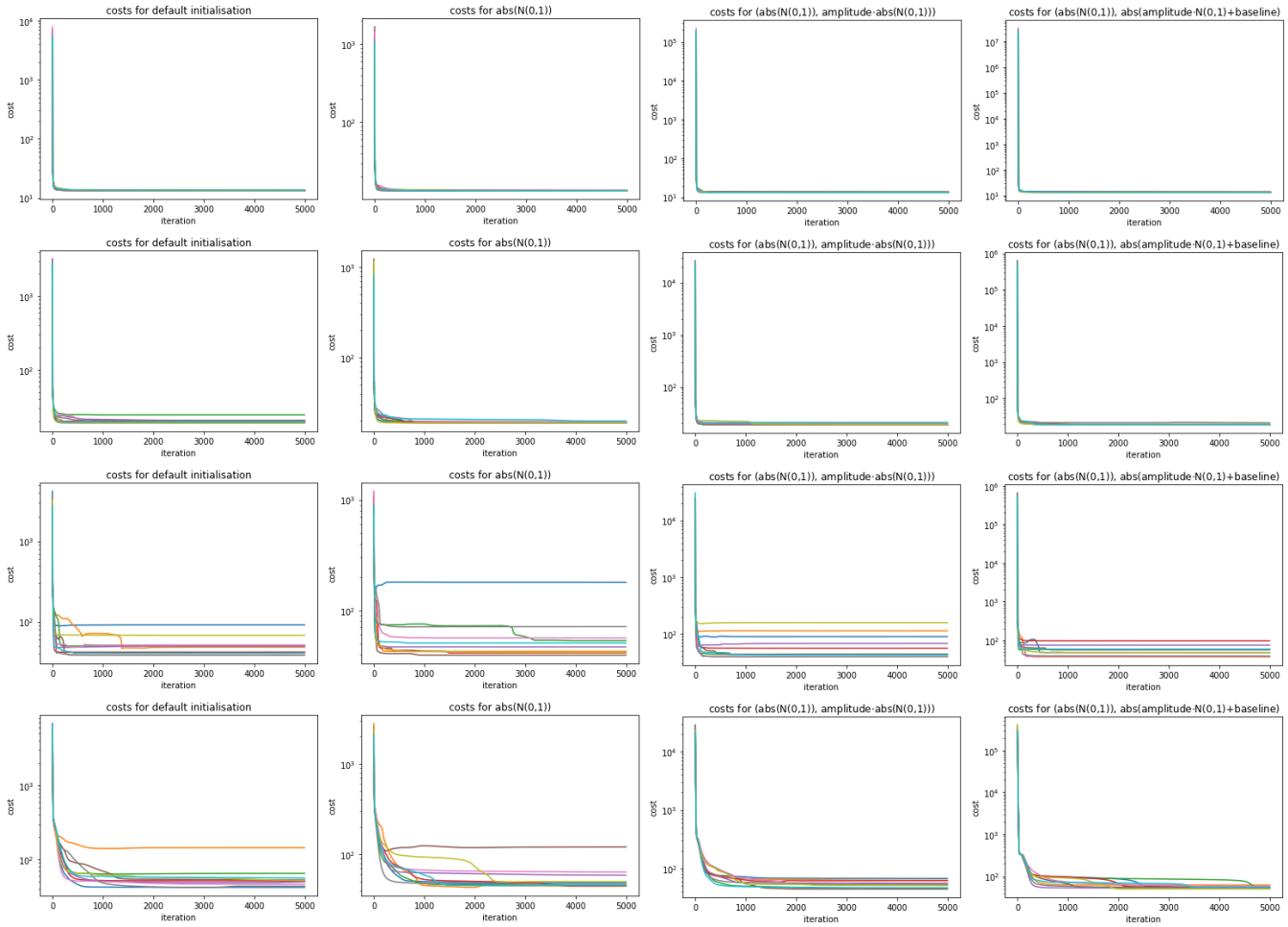


Figure 6. Cost at each iteration of 10 runs (each colour is a separate run) of each NMF algorithm for various initialization when factorising the same $10 \times 25$ matrix with coefficients distributed as $|\mathcal{N}(0,1)|$ and $K = 5$ for 5000 iterations. From top to bottom : EUC, KL, IS-MU, IS-EM. From left to right, coefficients initialized as : $|\mathcal{N}(0,1) + 1|$, $|\mathcal{N}(0,1)|$, $|\mathcal{N}(0,1)|$ for $W$ and $10|\mathcal{N}(0,1)|$ for $H$, $|\mathcal{N}(0,1)|$ for $W$ and $10|\mathcal{N}(0,1)| + 100$ for $H$.

out to be more sensitive to zero entries and converges to noisier components.

### C. Pitch estimation in the chords progression

KL proved to provide the best factorisation: for $K = 8$ all pitches were detected even though some of them were split in a few components (Fig.4). On the other hand IS-MU and IS-EM did not converge to a good factorisation: even after 5000 iterations the cost for IS-MU and IS-EM is higher than $10^7$. The resulting decompositions are very noisy and do not allow any musical interpretation. Finally EUC did not detect two out of four pitches even when $K$ was increased.

Even though EUC did not recover the pitches completely it can be used for denoising. It turns out that in its factorization the noise components are not present and obtained $WH$ can be treated as a denoised matrix $V$. Spectrogram for $WH$ for $K = 7$ is shown on Fig.5. It can be observed that amount of noise is significantly reduces when compared to the spectrogram on Fig.3. In audio restored from $WH$ all the chords are still present, but the noise of people talking in the background is replaced by somewhat more mechanical noise. We have also restored the recording for the KL factorization manually excluding noisy components. The resulting audio also has some mechanical noise on the background, but the chords sound much cleaner and sharper than in restoration from EUC factorization.

We have observed that increasing the $K$ value does not necessarily leads to detecting new pitches. For example, for EUC, even with $K = 8$, pitches 65 and 72 were not detected, but pitches 61 and 68 were split in two and four components respectively.

It is also interesting to notice that both EUC and KL detect a low frequency noise that is probably the sound from key attack.

## V. Conclusion

In summary we have seen that NMF gives us factorisation that have a meaningful interpretation in term of sound components as opposed to traditional methods such as PCA and SVD. Those factorisation allows us to identify and manipulate sound components independently of each other, with application ranging from pitch estimation (which could be extended to automatic transcription of an audio file to a score) to denoising.

We have shown that the choice of specific $\beta$-divergence as cost function can significantly affect the performance of NMF multiplicative gradient and EM algorithms both in factorisation and convergence to local minima, while also affecting which audio features were selected. In particular we have seen that EUC and KL performs better than IS-MU and IS-EM when the matrix to be factorised is not carefully conditioned, but that ensuring that every coefficients is strictly positive can vastly improve the performance of IS-MU and IS-EM and bring it on par with EUC and KL.

We have shown on some real audio recording how EUC and KL were able to recover some to all the pitches respectively, but that the pitches are split between components. We also observed the identification of some unexpected audio features such as key attack on the piano. We have shown that EUC in particular directly got rid of the noise and thus do not require truncation to obtain denoised reconstruction. We conclude that NMF is a useful tool, although computationally expensive, for sound feature extraction but that it needs careful choice of hyper-parameters and initialisation to reliably perform well.

## References

[1] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, mar 2009.

## APPENDIX A: DERIVATION OF ALGORITHM 1 FOR $\beta \in [1, 2]$

**Lemma 1.**
If $g(y, \tilde{y})$ for $D(y)$ then the update

$$y_{new} = \arg\min_y g(y, y_{old})$$

is non increasing for D(y)

*Proof of lemma 1:*

$$D(y_{new}) \leq g(y_{new}, y_n) \leq g(y_{old}, y_{old}) = D(y_{old}) \square$$

We show the derivation of the algorithm for the matrix $H$ only an analog proof can be done for the matrix $W$.

**Theorem**

- The function
  $g(H, \tilde{H}) = \frac{1}{\beta(\beta-1)} \sum_{f,n} V_{f,n}^{\beta}$
  $- \frac{1}{\beta-1} \sum_{f,k,n} V_{f,n} \frac{W_{f,k}\tilde{H}_{k,n}}{(W\tilde{H})_{f,n}} (\frac{(W\tilde{H})_{f,n}}{W_{f,k}\tilde{H}_{k,n}} W_{f,k}H_{k,n})^{\beta-1}$
  $+ \frac{1}{\beta} \sum_{f,k,n} \frac{H_{k,n}^{\beta}}{\tilde{H}_{k,n}^{\beta-1}} W_{f,k}(W\tilde{H})_{f,n}^{\beta-1}$
  is an auxiliary function for $D(H) := D(V|WH)$
- $\arg\min_H g(H, \tilde{H}) = \tilde{H}.\frac{W^T((W\tilde{H})^{\cdot\beta-2}.V)}{W^T(W\tilde{H})^{\cdot\beta-1}}$

*Proof of Theorem:*

- We can check that $g(H, H) = D(V|WH)$.
  We have to show that for all $H$ and $\tilde{H}$, $g(H, \tilde{H}) \geq D(H)$, for this purpose we want to find an appropriate lower bound for the second and third term of the function $g$. The following inequality for the second summands:

$$\sum_{f,k,n} V_{f,n} \frac{W_{f,k}\tilde{H}_{k,n}}{(W\tilde{H})_{f,n}} (\frac{(W\tilde{H})_{f,n}}{W_{f,k}\tilde{H}_{f,k}} W_{f,k}H_{k,n})^{\beta-1} \leq \sum_{f,k,n} V_{f,n}(W_{f,k}H_{k,n})^{\beta-1}$$

  holds by concavity of $x \to x^{\beta-1}$ for $\beta \in [1, 2]$
  Also for any $a_k$ and $b_k \in \mathbb{R}$ by convexity of $x \to x^{-\beta+1}$

$$\frac{\sum_k a_k b_k^{-\beta+1}}{\sum_k b_k} \geq (\frac{\sum_k a_k b_k}{\sum_k b_k})^{-\beta+1}$$

  Equivalently, for a fixed pair $(f, n)$, if we replace $a_k = \frac{H_{k,n}}{\tilde{H}_{k,n}}$ and $b_k = W_{f,k}\tilde{H}_{k,n}$ we get:

$$(\sum_k W_{f,k} \frac{H_{k,n}^{\beta}}{\tilde{H}_{k,n}^{\beta-1}})(\sum_k W_{f,k}\tilde{H}_{k,n}^{\beta-1}) \geq (\sum_k W_{f,k}H_{k,n})^{\beta}$$

  Since $W$, $H$ and $\tilde{H}$ are nonnegative we get that the third summands is bigger than $\frac{1}{\beta}(\sum_{f,k,n} W_{f,k}H_{k,n})^{\beta}$.

- $$\frac{\partial g(H, \tilde{H})}{\partial H_{k',f'}} = \beta(\frac{\tilde{H}_{k',f'}}{H_{k',f'}})^{2-\beta} \sum_n V_{n,f'}(W\tilde{H})_{n,f'}^{\beta-2} W_{n,k'} + \beta(\frac{\tilde{H}_{k',f'}}{H_{k',f'}})^{\beta} \sum_n W_{n,k'}(W\tilde{H})_{n,f'}^{\cdot\beta-1}$$

  Hence, by solving the 0 of the derivative and after simplification $\arg\min_H g(H, \tilde{H}) = \tilde{H}.\frac{W^T((W\tilde{H})^{\cdot\beta-2}.V)}{W^T(W\tilde{H})^{\cdot\beta-1}} \square$

## APPENDIX B: PROOF OF THEOREM 1.

$max_{W,H} p(X \mid W, H) \leftrightarrow min_{W,H} - log\ p(X \mid W, H) = -\sum_n log\ p(x_n \mid W, H) = -\sum_{f,n} \mathcal{N}(x_{fn} \mid 0, \sum_k w_{fk}h_{kn})$ where in the first equality we used the independence of each time frame and for the second equality the fact that $\mathcal{N}(0, h_{kn} \cdot diag(w_{:k})) \sim (\mathcal{N}(0, w_{fk}h_{kn}))_{f=1}^{F}$ since the covariance matrix is diagonal and that the sum of a (centered) gaussians is again a (centered) gaussian with variance the sum of their variances. Then using the density of gaussian and getting rid of some constants we get $min_{W,H} - \sum_{f,n} \mathcal{N}(x_{fn} \mid 0, \sum_k w_{fk}h_{kn}) \leftrightarrow min_{W,H} \sum_{f,n} [log(\sum_k w_{fk}h_{kn}) + \frac{|x_{fn}|^2}{\sum_k w_{fk}h_{kn}}] \leftrightarrow min_{W,H} \sum_{f,n} d_{IS}(|x_{fn}|^2 \mid (WH)_{fn}) = d_{IS}(|X|^{\cdot 2} \mid WH)$ where to obtain $d_{IS}$ we added some constants only depending on $x_{fn}$, and finally we used the definition of our cost function for $d_{IS}$ on matrices. $\square$

## APPENDIX C: DERIVATION OF ALGORITHM 2

**Expectation step:**
Do note that we are only considering $C_k$, a single component, so the previous $\sum_{k'} w_{fk'}h_{k'n} = w_{fk}h_{kn}$ in the next derivations. Let $Q_k(W_{:k}, H_{k:} \mid W^{(t-1)}, H^{(t-1)}) := E_{C_k}[-log\ p(C_k \mid W_{:k}, H_{k:}) \mid X, W^{(t-1)}, H^{(t-1)}]$. By the derivation in Appendix A, $-log\ p(C_k \mid W_{:k}, H_{k:}) = \sum_{f,n} [log(w_{fk}h_{kn}) + \frac{|(C_k)_{fn}|^2}{w_{fk}h_{kn}}]$ up to some constant.
Then note that $p((C_k)_{:n} \mid X, W^{(t-1)}, H^{(t-1)}) = \mathcal{N}((C_k)_{:n} \mid \mu_k^{(t-1)}, \Sigma_k^{(t-1)})$ with mean and covariance matrix estimates obtained using Wiener filtering, i.e., $\mu_k^{(t-1)} = \frac{W_{:k}H_{k:}}{WH}.X$ and $\Sigma_k^{(t-1)} = \frac{W_{:k}H_{k:}}{WH}.(WH - W_{:k}H_{k:})$. Then since it is a gaussian we can compute the second moment $(V_k^{(t-1)})_{fn} := E[|(C_k)_{fn}|^2] = |(\mu_k^{(t-1)})_{fn}|^2 + (\Sigma_k^{(t-1)})_{fn}$.
Combining the above we get: $min_{W,H} Q_k(W_{:k}, H_{k:} \mid W^{(t-1)}, H^{(t-1)}) \leftrightarrow min_{W,H} \sum_{f,n} [log(w_{fk}h_{kn}) + \frac{(V_k^{(t-1)})_{fn}}{w_{fk}h_{kn}}] \leftrightarrow min_{W,H} d_{IS}(V_k^{(t-1)} \mid WH) = \sum_{f,n} [\frac{(V_k^{(t-1)})_{fn}}{w_{fk}h_{kn}} - log(\frac{(V_k^{(t-1)})_{fn}}{w_{fk}h_{kn}}) - 1]$ by proceeding exactly as in Appendix A and using the linearity of the expectation.

**Maximisation step:**

$\frac{\partial}{\partial h_{kn}} d_{IS}(V_k^{(t-1)} \mid WH) = \sum_f [-\frac{(V_k^{(t-1)})_{fn}}{w_{fk}(h_{kn})^2} + \frac{1}{h_{kn}}] = \frac{F}{h_{kn}} - \frac{1}{(h_{kn})^2} \sum_f \frac{(V_k^{(t-1)})_{fn}}{w_{fk}}$. Setting to 0 we get $h_{kn}^{(t)} = \frac{1}{F} \sum_f \frac{(V_k^{(t-1)})_{fn}}{w_{fk}^{(t-1)}}$ if $h_{kn} > 0$. Similarly and using our new updated $h_{kn}^{(t)}$ we get $w_{fk}^{(t-1)} = \frac{1}{N} \sum_n \frac{(V_k^{(t-1)})_{fn}}{h_{kn}^{(t)}}$ if $w_{kn} > 0$. Using the above definition of $(V_k^{(t-1)})$ we get the updates in Algorithm 2. □