Non-Negative Matrix Factorization Report

EL GOUCH Hussein, ESSAADANI Obeye

Design

The practical was organised in four questions that build on each other: first derive the multiplicative update rules (MUR) from the β-divergence cost, then implement them, then experiment with parameters on a real audio signal (Cmajor_piano8khz.wav), and finally use the factorisation for source separation. We tried to keep the same logical thread in our notebook: every experiment answers a specific question, and every plot is followed by a short written observation. We also added two things that were not strictly required but that we found useful, a sanity check on synthetic rank-K data (to verify our implementation converges monotonically before applying it to real signals), and a Wiener-style reconstruction at the end, which is more natural for the Itakura–Saito divergence than a simple hard mask.

Implementation

The derivation in Question 1 was the most rewarding part on paper. The key idea is the non-negative split of the gradient into $\nabla^+ - \nabla^-$: once you see that both terms are non-negative because $W, H, X \geq 0$, the "magic" choice of step size $\eta = h/\nabla^+$ turns a gradient descent step into the multiplicative ratio in the subject's equations (3)–(4). The two updates for W and H are perfectly symmetric, which also gives a nice way to double-check the algebra.

For the code, we wrote a single unified update that works for any β, including $\beta = 2$: writing a special case for the Euclidean cost would just be repeating the general formula with $\beta - 2 = 0$. The β-divergence function itself, on the other hand, does need three branches because the formula in equation (2) has a $\beta(\beta - 1)$ in the denominator, which is singular at $\beta = 0$ and $\beta = 1$ — there the divergence has to be defined by its analytic limit (Itakura–Saito and Kullback–Leibler respectively).

Numerical robustness was the part we underestimated. The matrices $\hat{X} = WH$ appear with negative powers ($\hat{X}^{\beta-2}$ is $\hat{X}^{-2}$ when $\beta = 0$), so any zero or near-zero entry blows up the update. A small $\epsilon = 10^{-10}$ floor on W, H and $\hat{X}$ at every iteration solved it, but we had to be careful that the floor was applied after the multiplication, not before, otherwise the convergence stalls.

Difficulties encountered

The biggest surprise was the audio itself. We initially assumed the file was the C major scale starting at middle C (C4–C5), and we labelled our reconstructed components accordingly. After running NMF and listening to the components, the pitches did not match our labels at all. A YIN pitch-track on the original signal showed the scale is actually one octave lower (C3–C4) and, more importantly, the piece plays the scale ascending and then descending, not just ascending. That second fact completely changes how to read the activations H: each pitch should be active at two times in the recording, not one.

A subtler difficulty was identifying which component represents which pitch automatically. Our first attempt was "the lowest peak in W is the fundamental", but low piano notes routinely have a stronger 2nd or 3rd harmonic than their fundamental (a known consequence of soundboard

physics). So our peak-picker kept reporting the 3rd harmonic of C3 (≈ 392 Hz) and concluding it was "G4". In the end we gave up on automatic labelling and sorted the components by the time of their first significant activation in H, this is a much more reliable proxy here, since the ascending phase plays the eight pitches one after the other.

The choice of K was also instructive. With $K = 8$ (the number of distinct pitches), the algorithm occasionally allocates one component to a broadband attack-transient shared across all notes, leaving only seven proper notes split among the remaining slots, and one "missing" pitch. We observed that even slight changes to the random initialisation could swap which pitch got merged. With $K = 9$ the result was more stable across seeds, but then strictly speaking the "one component per pitch" interpretation no longer holds.

Reflection on the deeper challenges

Three challenges felt fundamental to us, not specific to our code:

- **Model selection.** NMF gives no automatic way to pick K. Too small and pitches merge; too large and they split into attack/sustain or absorb noise. In this practical we could choose K by knowing the answer (eight pitches), but in a real source-separation setting we wouldn't have that luxury.
- **The unsupervised nature.** Even with the right K, nothing in the cost function says "one component = one note". The factorisation that minimises the β-divergence may well be one where two notes are blended in one component. The interpretation is ours, not the algorithm's.
- **The STFT trade-off propagates.** Choosing a short window gives sharp note onsets but smears the harmonic comb that NMF needs to tell pitches apart; a long window does the opposite. NMF inherits whatever compromise the STFT makes, it cannot recover information that is not in the input.

The choice of β tied these together for us. With Euclidean $\beta = 2$, the cost is dominated by the few high-energy bins; faint harmonics, which are exactly what distinguishes one piano note from another, count for almost nothing. The Itakura–Saito divergence ($\beta = 0$) treats faint and loud bins on equal footing because it is scale-invariant. On this signal the difference was very visible in the W/H plots: with $\beta = 0$ each component cleanly captured one harmonic comb with two activation peaks, whereas with $\beta = 2$ several components overlapped. That this matches the maximum-likelihood interpretation under a Gaussian-STFT model is, we think, the single most elegant idea we take away from this practical.

Conclusion

We leave the practical with a working β-NMF implementation that we trust, a clear picture of why IS divergence is the right cost for audio power-spectrograms, and a healthy respect for how hard "unsupervised" really is. Most of our time was not spent on the math but on understanding the data and choosing reasonable hyper-parameters, which, in retrospect, is probably the right ratio.