Accelerated Nonnegative Matrix Factorization Methods for Percussion Source Separation

Aidan J. Beery [1]

Abstract

Hierarchical Least Squares (HALS) and Multiplicative Updates (MU) are two key nonnegative matrix factorization (NMF) solver algorithms. (Gillis & Glineur, 2012) present an accelerated method for these algorithms, sequentially updating each factor matrix before alternating to improve runtime performance. We implement these accelerated NMF algorithms for the task of audio source separation. Specifically, we use percussion audio and apply NMF to separate kick drum, snare drum, and hi-hat audio into separate tracks. We compare the runtime performance of A-HALS and A-MU and find that A-HALS converges to a better approximation of our source audio.

1. Introduction

Nonnegative Matrix Factorization (NMF) is a broadly applicable technique in signal processing. Given some input matrix $M \in \mathbb{R}_+^{m \times n}$, the objective is to find two component matrices, denoted $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that $M \approx WH$. Some integer $r \in \mathbb{Z}$ such that $r \leq min\{m, n\}$ is chosen to represent the rank of the factorization. We require M to be non-negative, meaning $M_{i,j} \geq 0 \quad \forall i < m, j < n$. To find these approximate factor matrices, the following optimization formulation is commonly used:

$$\min_{W \in \mathbb{R}_+^{m \times r}, H \in \mathbb{R}_+^{r \times n}} ||M - WH||_F^2 \;\; s.t. \;\; W \geq 0, H \geq 0$$

Nonnegative matrix factorization has been applied to a range of data science and scientific computation problems, including image analysis, dimensionality reduction, hyperspectral sensing, unsupervised clustering, topic modeling, and audio signal analysis (Fu et al., 2019) (Ozerov & Fevotte, 2010). There is an active interest in identifying fast, efficient numerical NMF solvers. Typically, these algorithms can be broadly classified as block coordinate descent methods, where one factor matrix is frozen as the other is updated, and this process alternates until some convergence criteria is met.

Algorithm 1 A general form for nonnegative least squares

Initialize $W^{(0)}, H^{(0)} \sim U(0, 1)$
for k *iterations* **do**
  Fix $H^{(k)}$
  Solve for $W^{(k+1)}$ such that:
  $||M - W^{(k+1)}H^{(k)}||_F^2 < ||M - W^{(k)}H^{(k)}||_F^2$

  Fix $W^{(k+1)}$
  Solve for $H^{(k+1)}$ such that:
  $||M - W^{(k+1)}H^{(k+1)}||_F^2 < ||M - W^{(k+1)}H^{(k)}||_F^2$
end

Like many other least-squares problems, nonnegative matrix factorization is solvable using alternating least-squares (ALS), so long as nonnegativity constraints on M, W, and H are met. To solve this least-squares formulation, it has been observed that each update for $W^{(k)}$ and $H^{(k)}$ can be computed iteratively (Cichocki et al., 2007). By fixing all but one column of W, there exists a closed-form solution to compute the optimal column $W(:, p)$ independently of the minimization of all other columns [1]. This method of sequentially updating each column of $W^{(k)}$ and $H^{(k)}$ is denoted hierarchical least-squares (HALS).

Another general non-negative least squares solver computes the update to each $W^{(k)}$ and $H^{(k)}$ using multiplicative updates (MU) (Berry et al., 2007). The Euclidian distance (and later, Frobenius norm) are shown to be monotonically nonincreasing given the multiplicative update rule $W^{(k)} \circ \frac{MH^{(k)T}}{W^{(k)}H^{(k)}H^{(k)T}}$. (Lee & Seung, 2000). Using this property, and alternating the update of $W^{(k)}$ and $H^{(k)}$ as done before, this multiplicative update algorithm will

[1]. Correspondence to: Aidan Beery <beerya@oregonstate.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

[1]I am unsure if HALS applies only to nonnegative problems, or can be used as a general least squares solver?

converge to minima for W and H. However, it has been shown that this convergence is relatively slow, and such multiplicative update methods have largely been superseded by HALS and projected gradient methods.

2. Accelerated Algorithms for NMF

Algorithm 2 MU update for $W^{(k)}$

$A = MH^{(k)T}$
$B = H^{(k)}H^{(k)T}$
$C = W^{(k)}B$
$W^{(k+1)} = W^{(k)} \circ \frac{A}{C}$

In their 2012 report, (Gillis & Glineur, 2012) present an accelerated form for HALS and MU. Typically, an alternating least squares algorithm will alternate between $W^{(k)}$ and $H^{(k)}$ after each update ($W^{(k+1)}$ and $H^{(k+1)}$, respectively). However, at each iteration of this alternating method (which they refer to as the 'outer iteration'), there is a fixed cost of recomputing matrices A and B (see algorithm 2). These matrices are invariant during each individual update, and A_H, B_H (the A, B computed for the update of $W^{(k)}$) only need to be recomputed after an update is made to H. The cost of recomputing A and B is the dominant computational cost of each outer iteration, as shown in (Gillis & Glineur, 2012).

To minimize the computational cost of both MU and HALS solvers, Gillis and Glineur propose an accelerated nonnegative least squares solver, where updates to $W^{(k)}, W^{(k+1)}, ..., W^{(k+\ell)}$ occur sequentially before an update to $H^{(k+1)}$ occurs. We denote these repeated independent updates as the function's inner loop, where $W^{(k,1)}, ..., W^{(k,\ell)}$ represents the ℓth inner update of the kth outer iteration. A and B must only be recomputed at each outer iteration k, and are fixed for each inner iteration $(k, 1), ...(k, \ell)$.

There is now the issue of identifying the optimal number of inner iterations ℓ at each update of $W^{(k)}$ and $H^{(k)}$, such that we minimize the number of outer iterations (and thus, computations of A and B). Gillis and Glineur propose two methods for estimating this optimal number of inner updates. One, a 'fixed' method, uses the estimated runtime cost of each update to determine a stopping criteria for this inner loop. [2] Alternatively, the interior loop can simply iterate until the distance between the previous update and the current one is within some threshold ϵ relative to the distance of the first interior update, at which point the interior update loop ends. This dynamic criteria is defined formally by $||W^{(k,\ell+1)} - W^{(k,\ell)}||_F \leq \epsilon||W^{(k,1)} - W^{(k,0)}||_F$.

[2]Note that, rather than estimate this cost, we directly measure the runtime of an update of A and B in our implementation.

Algorithm 3 Accelerated Coordinate Block Descent

Data: $M \in \mathbb{R}_+^{m \times n}$, $(W^{(0)}, H^{(0)})$
for $k = 0, 1, 2, ...$ **do**
  Start timer t
  $A = MH^{(k)T}$
  $B = H^{(k)}H^{(k)T}$
  $t \Leftarrow Elapsedtime$
  while *runtime* $> \alpha t$ **do**
    Compute $W^{(k,\ell)}$ using block update method
    if $||W^{(k,\ell+1)} - W^{(k,\ell)}||_F \leq \epsilon||W^{(k,1)} - W^{(k,0)}||_F$ **then**
      break
    end
    ℓ++
  end
  $W^{(k+1)} = W^{(k,\ell)}$ Start timer t
  $A = W^{(k+1)T}M$
  $B = W^{(k+1)T}W^{(k+1)}$
  $t \Leftarrow Elapsedtime$
  while *runtime* $> \alpha t$ **do**
    Compute $H^{(k,\ell)}$ using block update method
    if $||H^{(k,\ell+1)} - H^{(k,\ell)}||_F \leq \epsilon||H^{(k,1)} - H^{(k,0)}||_F$ **then**
      break
    end
    ℓ++
  end
  $H^{(k+1)} = H^{(k,\ell)}$
end

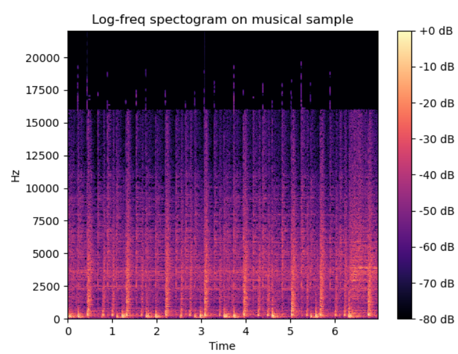
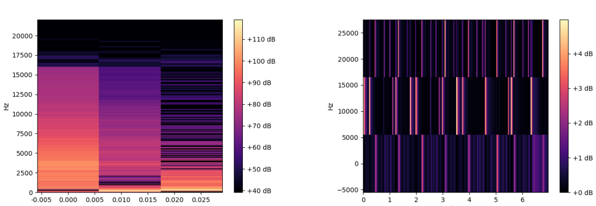


Figure 1. Spectogram of percussion acoustic sample



(a) Factor matrix W (b) Factor matrix H

Figure 2. Low-rank decomposition of percussion musical sample

3. A-HALS & A-MU for Source Separation

Nonnegative matrix factorization is broadly applicable to a range of music signal processing tasks (López-Serrano et al., 2019), such as acoustic feature extraction, voice detection, and rhythm structure analysis. We focus on the application of NMF to source separation. Given a mixed audio sample, our goal is to extract each instrumental track as a separate waveform. Commonly, access to the original multitrack audio sample is only available to the music producer who performed the original audio mixing. However, for music production tasks such as remixing or audio restoration, these multitrack audio files are necessary for editing work.

Our goal is to apply accelerated HALS and accelerated MU to separate instrument tracks from a given audio sample. A hypothetical user would typically be running source separation from a digital audio workstation (DAW), and thus would benefit greatly from a tool which is efficient enough to be ran on a typical desktop computer unlike computationally expensive neural methods (Hennequin et al., 2020). We experiment with four separate percussion audio tracks, and present spectrograms and waveform results for one of these tracks in this report. Additional materials, including recordings of mixed audio and separated instrumental tracks, can be found on our webpage here: https://aidanbeery.com/portfolio/08nmf-music/.

We first convert our audio samples to spectrograms using a short-time Fourier transform implemented in Librosa [3], an open-source audio processing framework. From this, we take the elementwise absolute value of each element of this spectrogram, yielding our amplitude spectrogram (see Figure 1). By using the amplitude spectrogram, our input meets the nonnegativity condition and we can use NMF algorithms to find a low-rank representation of this acoustic signal.

We follow the training paradigm outlined in algorithm 3, using both fixed and dynamic stopping criteria for our inner loops. The number of components of our factor matrices (r) is determined manually by the observed number of instruments in the audio sample. This parameter depends on the goal of the source separation. For example, if separating a multi-instrumental track, r would be the number of instruments. However, if the goal is to separate a vocal track from a mixed instrumental track (a common task in audio post-production), then r would be set to 2.

Our sample track in Figure 1) is a percussion solo, and our goal is to separate the kick drum, snare drum, and cymbals in the audio track. In such a case, our goal is to extract $r = 3$ components. We apply both Accelerated HALS and Accelerated MU, using the parameters defined in (Gillis & Glineur, 2012) ($\alpha = 0.5$, $\epsilon = 0.1$ for A-HALS, $\alpha = 2, \epsilon = 0.1$ for A-MU). For each algorithm, we run 300 outer iterations.

[3] https://librosa.org/

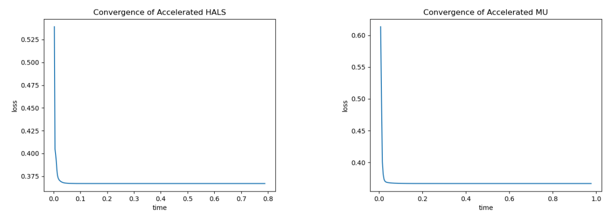


Figure 3. Convergence of A-HALS and A-MU

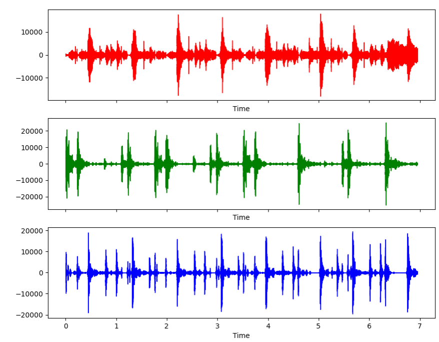


Figure 4. Separated waveforms, after inverse STFT

For each component r, we take the outer product of the corresponding row of H and column of W to yield a source spectrogram (see Figure 5). We then apply an inverse short-time Fourier transform to each component spectrogram, yielding the source waveforms shown in Figure 4.

Comparing the runtime performance of A-MU and A-HALS, our results are consistent with those found in (Gillis & Glineur, 2012) and the broader NMF literature. We measure loss as $||M - WH||_F^2$. We report time as CPU runtime [4] in place of number of iterations, as the number of inner iterations vary per sample depending on our stopping criteria. HALS converges faster than our multiplicative update based method, and appears to converge to a better approximation.

4. Conclusion

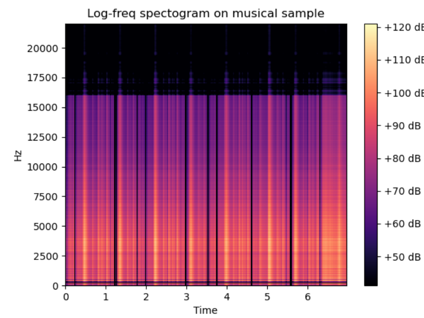
Nonnegative matrix factorization can be a robust, reliable tool for audio signal processing. Unlike contemporary deep learning approaches, NMF-based source separation programs are computationally efficient, allowing for deployment on resource-constrained devices and creating the potential for accelerating creative workflows on mobile devices.

[4] Using Python's `time.perf_counter()`

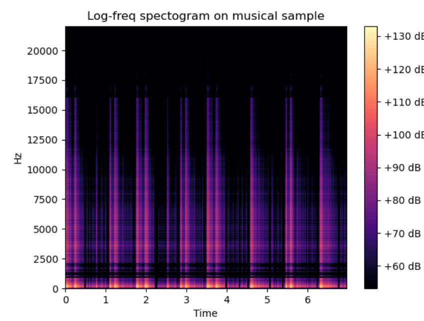
Gillis and Glineur's method for accelerating coordinate block descent algorithms for nonnegative least-squares problems is simple and straightforward to implement. In practical terms, this accelerated method can 'wrap' an existing alternating least-squares algorithm, simply requiring the update functions be implemented with a maximum time threshold and a early stopping criteria based on the distance at each update. NMF solvers using multiplicative update rules are simple to understand and implement, however they suffer from worse overall convergence than the more complex hierarchical least-squares algorithm. Both algorithms are efficient approaches to nonnegative matrix factorization [5].

4.1. Future work

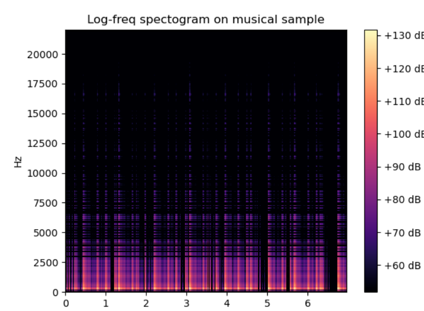
Source separation is an intrinsically noisy process, and yet its primary applications, such as audio production, require very low-noise outputs. Towards this, there has been ample work on filtering source audio (Le Roux & Vincent, 2012). In these experiments, we briefly tested the use of Wiener filtering as a post-processing step for our source separation program. However, naively applying a Wiener filter [6] with varying window sizes yields corrupted outputs. Existing community open-source implementations [7] using expectation-maximization yield valid waveforms, yet the generated samples were far too quiet and seemed to have some distortion. Future iterations of this program should more thoroughly investigate denoising strategies for the output source audio.



(a) Source 1: Kick Drum



(b) Source 2: Snare Drum



(c) Source 3: Cymbals

Figure 5. Spectograms of source components

References

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics Data Analysis*, 52(1):155–173, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.11.006.

Cichocki, A., Zdunek, R., and Amari, S.-i. Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization. volume 4666, pp. 169–176, 01 2007. ISBN 978-3-540-74493-1. doi: 10.1007/978-3-540-74494-8_22.

Fu, X., Huang, K., Sidiropoulos, N. D., and Ma, W.-K. Nonnegative matrix factorization for signal and data analytics:

[5] An outstanding question from my investigation: Is there any advantage to a MU-based solver, rather than a HALS-based one, besides simplicity? The HALS solver seems to converge faster and to a better solution without tradeoff.

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.wiener.html

[7] https://github.com/sigsep/norbert

Identifiability, algorithms, and applications. *IEEE Signal Process. Mag.*, 36(2):59–80, 2019.

Gillis, N. and Glineur, F. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural Computation*, 24(4):1085–1105, April 2012. ISSN 0899-7667, 1530-888X. doi: 10.1162/NECO_a_00256. arXiv:1107.5194 [math].

Hennequin, R., Khlif, A., Voituret, F., and Moussallam, M. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020. doi: 10.21105/joss.02154. URL https://doi.org/10.21105/joss.02154.

Le Roux, J. and Vincent, E. Consistent wiener filtering for audio source separation. *IEEE signal processing letters*, 20(3):217–220, 2012.

Lee, D. and Seung, H. S. Algorithms for non-negative matrix factorization. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

López-Serrano, P., Dittmar, C., Özer, Y., and Müller, M. Nmf toolbox: Music processing applications of nonnegative matrix factorization. In *Proceedings of the International Conference on Digital Audio Effects DAFx*, volume 19, pp. 2–6, 2019.

Ozerov, A. and Fevotte, C. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2010. doi: 10.1109/TASL.2009.2031510.