A Provably Correct and Robust Algorithm for Convolutive Nonnegative Matrix Factorization

Anthony Degleris* and Nicolas Gillis†

***Abstract*—In this paper, we propose a provably correct algorithm for convolutive nonnegative matrix factorization (CNMF) under separability assumptions. CNMF is a convolutive variant of nonnegative matrix factorization (NMF), which functions as an NMF with additional sequential structure. This model is useful in a number of applications, such as audio source separation and neural sequence identification. While a number of heuristic algorithms have been proposed to solve CNMF, to the best of our knowledge no provably correct algorithms have been developed. We present an algorithm that takes advantage of the NMF model underlying CNMF and exploits existing algorithms for separable NMF to provably find the unique solution (up to permutation and scaling) under separability-like conditions. Our approach guarantees the solution in low noise settings, and runs in polynomial time. We illustrate its effectiveness on synthetic datasets, and on a singing bird audio sequence.**

***Index Terms*—convolutive nonnegative matrix factorization, algorithms, robustness to noise, separability**

I. Introduction

Nonnegative matrix factorization (NMF) is a standard unsupervised learning technique for analyzing large datasets. Given an $N \times T$ matrix X, NMF seeks a $N \times K$ matrix $W \geq 0$ (where the inequality is elementwise) and a $K \times T$ matrix $H \geq 0$ such that $X \approx WH$ and $K \ll \min(N,T)$. NMF has been successfully applied to a number of practical problems; these include hyperspectral unmixing, text mining, audio source separation, and image processing; see [1], [2], [3] and the references therein. Below, we consider three important limitations of NMF, each of which motivate this paper.

A first limitation of NMF is that it fails to capture local dependencies in the data; in other words, permutations of the columns of the input matrix do not affect NMF (beyond the permutation of the columns of H). For example, in imaging applications a column of X may represent a pixel, and adjacent columns will often correspond to pixels adjacent to one another in the image. Neighboring pixels tend to be quite similar, especially in low contrast images. In audio or neuroscience datasets, each column is a certain instant in time, and therefore neighboring columns are often highly interdependent. To capture these local dependencies, Smaragdis [4] proposed a convolutive variant of NMF, known as convolutive NMF (CNMF). CNMF attempts to find L matrices $W_1, \dots, W_L$ of size $N \times K$ and a matrix H of size $K \times T$ such that $X = \sum_{\ell=1}^{L} W_\ell H S_{\ell-1}$, where S_τ is a square matrix that shifts the columns of H by τ places to the right, zero-padding the leftmost τ columns. Explicitly, S_τ has ones on its τth upper diagonal and zeros elsewhere. It is often convenient to instead define an $L \times N \times K$ tensor W and denote $W_{\ell ::} = W_\ell$ the resulting $N \times K$ matrix when the first index is fixed to ℓ. Then another, perhaps more intuitive, definition of CNMF is to equivalently write $X = \sum_{k=1}^{K} W_{::k}^T * h_k$, where $*$ is the 2D-convolution operator defined as

$$(A * b)_{ij} = \sum_{\tau=1}^{\max(L,j)} A_{i\tau} b_{j+1-\tau} \in \mathbf{R}^{N \times T}$$

where $A \in \mathbf{R}^{N \times L}$ and $b \in \mathbf{R}^{1 \times T}$, $W_{::k} \in \mathbf{R}^{L \times N}$ is the resulting matrix when the third index is fixed to k, and $h_k \in \mathbf{R}^{1 \times T}$ is the kth row of H. Thus, CNMF is a factorization of X into a sum of 2D-convolutions. Since its conception, CNMF has been found particularly useful for audio source separation [4], [5], [6] and neural sequence identification [7]. In general, CNMF will provide a more concise and interpretable factorization than NMF whenever the columns of X exhibit local dependencies, which is often the case when the columns represent points in time or space.

A second limitation of NMF is that solving NMF problems is challenging in practice. In fact, it is in general

* A. Degleris is with the Department of Electrical Engineering, Stanford University, Stanford CA, USA. E-mail: degleris@stanford.edu.

† N. Gillis is with the Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, Rue de Houdain 9, 7000 Mons, Belgium. NG acknowledges the support by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) under EOS Project no O005318F-RG47, and by the European Research Council (ERC starting grant no 679515). E-mail: nicolas.gillis@umons.ac.be.

Manuscript received in June 2019.

NP-hard [8], leading researchers to rely on a number of heuristic algorithms; see, e.g., [2] and the references therein. As CNMF is a generalization of NMF (the model is exactly NMF when $L = 1$), it inherits problems related to computational intractability. The first algorithm proposed was an averaged multiplicative update rule [4], which treats each pair $(W_\ell, HS_{\ell-1})$ as an NMF and updates them using an NMF multiplicative update [9], averaging the updates for H. Since then, numerous other algorithms have been proposed to fit CNMF, including true multiplicative updates [5], [10], [11], [12] (which are derived using majorization minimization), projected alternating least squares [13], alternating nonnegative least squares (ANLS; [14]), and hierarchical alternating least squares (HALS; [14]). Many of these algorithms are generalizations of algorithms for NMF and perform well on both synthetic and experimental data. Several of them also have convergence guarantees to stationary points. However, all the aforementioned algorithms are heuristics, in the sense that there is no guarantee that they reach a global minimum.

A third limitation of NMF is that, in general, the NMF WH of a matrix X is not essentially unique, that is, there may be other W', H' (of the same size as W, H) such that $W'H' = X = W'H'$ but $W' \neq WD\Pi$ and $H' \neq \Pi^{-1}D^{-1}H$ for any diagonal matrix $D \in \mathbf{R}^{K \times K}$ and permutation matrix $\Pi \in \{0,1\}^{K \times K}$. We therefore cannot be certain that the recovered factors (W, H) correspond to the true sources that generated the data (up to permutation and scaling). The same problem arises with CNMF; see Section V-A for some examples. In practice, most researchers use additional regularization terms to promote some structure in the sought solution such as sparsity [15]. Unfortunately, identifiability results for NMF are rather scarce; see the recent survey [3]. As far as we know, the only NMF model that is both *tractable* and *identifiable* is separable NMF [16]. When trying to identify an NMF $WH = X$, separable NMF makes the additional assumption that the columns of W are contained somewhere in X, that is, $W = X[:, \mathcal{C}]$ for some index set $\mathcal{C}$. In other words, if the matrix X has the form $X = X[:, \mathcal{C}]H$ for some index set $\mathcal{C}$ and $H \geq 0$ where $\mathcal{C}$ has minimal size (that is, there does not exist another index set $\mathcal{C}'$ of smaller cardinality such that $X = X[:, \mathcal{C}']H'$ for $H' \geq 0$), then any other separable NMF is the same, up to permutation and scaling. Geometrically, the columns of $X[:, \mathcal{C}]$ are the extreme rays of the cone generated by the columns of X; these extreme rays are unique (up to scaling) and have to be identified since, by definition of extreme rays of a cone, they cannot be reconstructed via a nonnegative linear combination of other points from the cone. This assumption is called the *separability assumption*. This terminology was introduced in the paper of Donoho and Stodden [17] where it was shown that the matrix X has an essentially unique NMF decomposition under the separability assumption *and* additional sparsity conditions. However, the separable NMF model dates back to the 1990's in the hyperspectral imaging literature where the separability assumption is referred to as the pure-pixel assumption; see [18] and the references therein. More recently, Arora et al. [16] proved that separable NMF model is essentially unique, robust to noise, and computable in polynomial time. Since then, numerous separable NMF algorithms, which aim to recover the index set $\mathcal{C}$ such that $X \approx X[:, \mathcal{C}]H$, have been proposed; see [19] and the references therein. This model has been used successfully in applications such as topic modeling [20], community detection [21], and the analysis of time-resolved Raman spectra [22], to cite a few. Nevertheless, the separability assumption is rather strong and does not hold in all applications. In these situations, separable NMF is still a powerful way to initialize more sophisticated NMF models, such as minimum-volume NMF models, which are identifiable but suffer from intractability [3]. One of the most popular and powerful separable NMF algorithms is the successive projection algorithm (SPA) [23] which is guaranteed to recover the columns of W, even in the presence of noise [24]. SPA is a greedy algorithm that sequentially identifies the columns of W using successive orthogonal projections (see Algorithm 3 and Appendix A for more details.). SPA is closely related to the modified Gram-Schmidt algorithm with column pivoting, and has rediscovered many times [2].

A. Contribution and Outline of the Paper

In this paper, we make two contributions. First, we establish conditions under which the CNMF problem has a unique solution (up to permutation and scaling of the rank-one factors). These conditions generalize the separability assumptions discussed in the previous subsection. Second, we propose a polynomial time algorithm that provably solves the CNMF problem under these conditions, even in the presence of small noise. This algorithm reduces CNMF to NMF with linear constraints and takes advantage of existing separability-based methods for NMF. In particular, we utilize SPA to estimate the columns of $W_1, \ldots, W_\ell$, then apply estimation, clustering, and sorting techniques to uncover the convolutive structure. We later generalize our approach to use any separable NMF algorithm and show how the choice of the separable NMF algorithm affects our assumptions, run-time, and noise tolerance. Additionally, we show that, like separable NMF algorithms, our method can be used as an efficient initialization strategy for heuristic CNMF algorithms. To the best of our knowledge, this is the first algorithm to provably solve the CNMF problem in either the absence or presence of noise. Hence, under separability-like assumptions, our approach resolves the three important limitations of NMF mentioned in the introduction.

The paper is organized as follows. In Section II, we state the CNMF problem and formally define the convolutive separability assumptions. In Section III, we propose the Locate-Estimate-Cluster-Sort Algorithm (LECS) for recovery and show that LECS provably finds the solution to the CNMF problem under the convolutive separability assumptions. In particular, Theorem 2 guarantees that LECS will find the unique optimal solution (up to permutation and scaling) in the absence of noise. Theorem 3 generalizes this result to problems with bounded noise. The proof of Theorem 3 is deferred to the Appendix.

Notation: The jth entry of a vector a is denoted by a_j. For matrices $A \in \mathbf{R}^{m\times n}$, we use $A[i,:] = A_{i:}$ and $A[:,j] = A_{:j}$ to denote the ith row of A and jth column of A, respectively. We will also use lower case letters for columns and rows, but define them explicitly first. The entry in the ith row and the jth column of A will be denoted $A[i,j] = A_{ij}$. A nonnegative vector a or a nonnegative matrix A are denoted using $a \in \mathbf{R}^n_+$ and $A \in \mathbf{R}^{m\times n}_+$. Define S_τ as a sqaure T-by-T matrix with 1 on its τth upper diagonal and zeros elsewhere. For vectors $a \in \mathbf{R}^n$, the p-norm for $1 \leq p < \infty$ is defined as $\|a\|_p = \left(\sum_{j=1}^n |a_j|^p\right)^{1/p}$. The maximum and minimum p-norms of a column of A are denoted by $\|A\|_{p,col} = \max_{1\leq j\leq n} \|A_{:j}\|_p$ and $\|A\|_{-p,col} = \min_{1\leq j\leq n} \|A_{:j}\|_p$, respectively. We also define $\|A\|_{p,row} = \|A^T\|_{p,col}$ and $\|A\|_{-p,row} = \|A^T\|_{-p,col}$. The Frobenius norm is denoted as $\|A\|_F = \left(\sum_{i,j=1}^{m,n} A_{ij}^2\right)^{1/2}$. We denote $\sigma_{\max}(A)$ to be the largest singular value of A, and $\sigma_{\min}(A)$ to be the smallest singular value of A. The condition number of a matrix A induced by the 2-norm is denoted as $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$. Given a matrix $A \in \mathbf{R}^{m\times n}$, the diagonal matrix $D_A \in \mathbf{R}^{n\times n}$ is defined by $(D_A)_{ii} = \begin{cases} 0 & \text{if } \|A_{:i}\|_1 = 0 \\ \|A_{:i}\|_1^{-1} & \text{otherwise} \end{cases}$.

II. Problem Setup and the Separability Conditions

Suppose there is a matrix $X \in \mathbf{R}^{N\times T}_+$ generated as $X = \sum_{\ell=1}^L W_\ell H S_{\ell-1}$, where $W_\ell \in \mathbf{R}^{N\times K}_+$ for $\ell = 1,2,\ldots,L$, and $H \in \mathbf{R}^{K\times T}_+$. Consider $\tilde{X} = X + E$, where E is some matrix of noise. Given $\tilde{X}, K, L$, the *convolutive NMF problem* (or CNMF problem) is to approximately recover W_ℓ, H (up to scaling or permutation). When $E = 0$ and hence $\tilde{X} = X$, this problem amounts to finding an exact convolutive NMF.

A. Reformulation as Constrained NMF

Our approach will take advantage of existing literature on NMF. In particular, we look to separable NMF algorithms and attempt to extend them to the convolutive case. In this vein, it is useful to reformulate the convolutive NMF model as an NMF model with linear constraints of H.

Consider the sum $\sum_{\ell=1}^L W_\ell H S_{\ell-1}$. Define $H_\ell = H S_{\ell-1}$ for $\ell = 1,2,\ldots,L$. Now define

$$V = \begin{bmatrix} W_1 & \ldots & W_L \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} H_1 \\ \vdots \\ H_L \end{bmatrix}. \qquad (1)$$

It follows that $VG = \sum_{\ell=1}^L W_\ell H S_{\ell-1} = X$. Now we may think of the exact CNMF problem as a restriction of the exact NMF problem; Given X, K, L, our goal is to find V and G such that $VG = X$ and G is defined as in (1), that is, each block H_ℓ is a shifted version of the first block H_1. The above formulation further elucidates the relationship between NMF and CNMF. When $L = 1$, CNMF reduces to NMF. On the other hand, any CNMF (given by $X, W_1, \ldots, W_\ell, H$) is also an NMF (given by X, V, G). Therefore, when the V, G found from NMF have the form in (1), a CNMF has also been identified. Given the reformulation of (1), our approach is to utilize separable NMF algorithms to locate the columns of V within $\tilde{X}$ and then estimate G. Once these matrices have been identified up to permutation and scaling, we use clustering and sorting methods to identify H and $W_1, \ldots, W_L$. We name this approach the Locate-Estimate-Cluster-Sort Algorithm (LECS); see Algorithm 1 which is described in details in Section III.

B. Convolutive Separability

To introduce a notion of separability for the convolutive NMF model, we need to address two relevant details that are not present in the standard NMF model. First, to handle time dependencies, we need some notion of distance up to a shift. The following definition formalizes this notion mathematically.

Definition 1 (L-shift similarity)**.** Consider two vectors $g_i, g_j \in \mathbf{R}^T$. The L-shift cosine similarity $\cos_L$ between g_i and g_j is denoted $\cos_L(g_i, g_j)$ and given by

$$\max_{\ell=0,\ldots,L-1} \max\left(\cos\left(S_\ell^T g_i, g_j\right),\ \cos\left(g_i, S_\ell^T g_j\right)\right),$$

where $\cos(x, y) = (x^T y)/(\|x\|_2\|y\|_2)$. This is exactly the cosine of the minimum angle between g_i, g_j over all shifts of length $\ell < L$.

The utility of the L-shift similarity comes from the fact that $\cos_L(x, y) = 1$ is equivalent to the statement that either $x = \alpha S_\ell^T y$ or $y = \alpha S_\ell^T x$ for some $\ell \in \{0, \ldots, L-1\}$ and for some scalar $\alpha > 0$.

The second detail is minor but essential. Separable NMF algorithms like SPA require that each column of X lie in the convex hull of V, that is, $\sum_i G[i, j] \leq 1$ for all j (see Appendix A for more details). Nevertheless, when this does not hold we can still scale the columns of V and X to sum to one, and thus apply SPA. This is not the case in the convolutive NMF model, and hence we must develop a variant of SPA that can handle separable NMF without this condition (see Algorithm 3). . Since the magnitude of the columns of $W_{::k}$ may change over time (for example, a crescendo in an audio sequence), the columns of V may not have a constant sum. Thus, we will have to allow arbitrary positive scaling of V by some positive diagonal matrix A, and apply SPA to VA. With these details in mind, we can now formulate a concept of separability for the CNMF problem. We first provide a definition, then explain each condition in more details.

Definition 2 (Convolutive separable)**.** Given a CNMF problem with inputs $\tilde{X} = X + E$ where $X = \sum_{\ell=1}^L W_\ell H S_{\ell-1}$, L and K, we say the problem is *convolutive separable* with respect to δ, ϵ, A, where $\delta > 0$, $\epsilon \geq 0$, and $A \in \mathbf{R}^{KL \times KL}$ is a diagonal matrix with strictly positive diagonal entries, if the following conditions are satisfied:

(A) **[Separable]** For each k, ℓ, the vector $W_k[:, \ell]$ appears as a scaled column of X. Explicitly, there are several equivalent ways to express this condition.

- The matrix V satisfies $VA = X[:, \mathcal{C}]$ for some set $\mathcal{C}$.

- The matrix G satisfies $G = [A\ M]\Pi$ for some $M \in \mathbf{R}_+^{KL \times T-KL}$ and permutation matrix $\Pi \in \mathbf{R}^{T \times T}$.

(B) **[Sequentially unique]**

- For any two rows h_i, h_j of H, the $2L$-shift similarity between them is $\cos_{2L}(h_i, h_j) \leq 1 - \delta$.

- For any row h_i of H and any $\ell, \tau = 1, \ldots, L-1$, we have $\cos(S_\tau^T h_i, S_\ell^T h_i) \leq 1 - \delta$ when $\tau \neq \ell$.

(C) **[Full Rank]** The matrix V is full rank, that is, V has rank KL.

(D) **[Bounded noise]** The noise matrix E satisfies $\|E\|_{1,col} \leq \epsilon$.

Definition 2 guarantees a unique solution in the following sense.

Theorem 1 (Unique Solution)**.** *Suppose $X = \sum_{\ell=1}^L W_\ell H S_{\ell-1}$ satisfies (A), (B), and (C). Then any other CNMF given by $W_1', \ldots, W_L', H'$ that satisfies (A), (B), (C), and $X = \sum_{\ell=1}^L W_\ell' H' S_{\ell-1}$ differs at most by permutation and scaling, in the sense that $H = \Pi\Lambda H'$ for a positive diagonal matrix $\Lambda \in \mathbf{R}^{K \times K}$ and a permutation matrix $\Pi \in \mathbf{R}^{K \times K}$.*

The proof of this result follows directly from Theorem 2, since our proposed algorithm (namely, LECS described in Algorithm 1) is deterministic and guaranteed to recover the underlying CNMF up to permutation and scaling. The same approach entails the uniqueness of a solution to a noisy CNMF problem $\tilde{X} = \sum_{\ell=1}^L W_\ell H S_{\ell-1} + E$ among solutions satisfying the conditions of Theorem 3.

The first condition **[Separable]** is inherited directly from the separable NMF literature and used only to guarantee that a separable NMF algorithm can identify the columns of V. One exception is that in separable NMF problems, one usually requires that $V = X[:, \mathcal{C}]$. In this paper, we relax this condition to $VA = X[:, \mathcal{C}]$ for some diagonal matrix A with strictly positive diagonal entries. This weaker constraint allows the columns of V to be arbitrarily scaled in the matrix X. Ultimately, we use a separable NMF algorithm to identify the columns of VA and use this scaled matrix throughout the rest of our procedure. Some minor difficulties arise from this, which are addressed in Appendix A. In Section IV, we show that **[Separable]** can actually be weakened to the more general *sufficiently scattered* condition.

The second condition, **[Sequentially unique]** tells us that each row of H must differ from all other rows and their shifted variants. Moreover, it tells us the rows must be distinct from its own shifted variants. Since we are generally unconcerned with the scale of a row of H, the cosine of the angle between two vectors serves as a useful scale-invariant measurement of similarity (if the vectors have mean zero, it is exactly the correlation between the two vectors). **[Sequentially unique]** will be an important condition after we have a permuted estimate of G, since we will need to rearrange the rows to obtain a matrix that satisfies the constraints of (1). In practice, this is a very reasonable assumption; all we require is (a) that any two factors h_i, h_j must not be highly correlated (in which case we should just have one factor), and (b) that the factor h_j is not periodic with a period smaller than L (in which case we can choose a smaller L).

In our approach, **[Full Rank]** guarantees that SPA terminates successfully and locates the columns of V. We also use it to bound the error when estimating G via nonnegative least squares. However, other algorithms like the Successive Nonnegative Projection Algorithm (SNPA) from [25] do not require **[Full Rank]**. Similarly, [26] defines least squares robustness results in the rank deficient case. Although we assume **[Full Rank]** in our results to simplify our analysis, it can be weakened to the condition that no column of V is contained in the convex cone generated by the other columns of V. We elaborate on this in Section IV.

[Bounded noise] bounds the 1-norm of any column of the noise matrix E. When $E = 0$, this is clearly satisfied. Otherwise, we will later see that if ϵ is sufficiently small, we can still recover noisy estimates of $W_1, \dots, W_\ell, H$. Note that this bound differs slightly from the noise bounds in many separable NMF algorithms. In particular, most separable NMF algorithms require bounds on the 2-norm of any column of the noise matrix E, whereas we bound the 1-norm. Since $\|v\|_2 \leq \|v\|_1$ for any vector $v \in \mathbf{R}^n$, we make a stronger assumption on the noise. This stronger condition is intimately related to the following. In separable NMF, we require $\|H\|_{1,col} \leq 1$. In practical applications where this might not hold, we scale the columns of X so that this assumption is satisfied and show that the new problem is equivalent; see for example [24]. This scaling trick does not apply to CNMF however, since the matrix H is difficult to appropriately scale in the expression $\sum_{\ell=1}^{L} W_\ell H S_{\ell-1}$. In Appendix A, we explain this scaling trick and its limitations in more details.

Is convolutive separability reasonable in practice? In challenging real-world scenarios, the convolutive separability condition will most likely be violated, because several sources will overlap in all time windows. Also, the noise level will be typically higher than what our bounds allow (see Theorem 3). These two issues are analogous to those encountered by separable NMF. Nevertheless, as explained in the introduction, separable NMF has been used successful in many challenging real-world problems, either directly or as an initialization strategy. Accordingly, we believe many real-world signals can be approximated well under the convolutive separability assumption because, although several sources are active in all time windows, some time windows will contain a single source that is dominant. Moreover, as it will be shown in the experiments, our approach leads to efficient initialization strategies in challenging scenarios. Additionally, our approach can be generalized in a number of different ways (see Section IV). For example, the separability assumption can be relaxed to the sufficiently scattered condition by using minimum-volume NMF instead of separable NMF.

III. Algorithm Description and Recovery Guarantee

In this section, we first describe in details our proposed algorithm (LECS, Algorithm 1) in Section III-A, and then prove its correctness under the convolutive separability assumptions in the noiseless case (Section III-B) and in the presence of noise in (Section III-C).

Algorithm 1 Locate-Estimate-Cluster-Sort Algorithm (LECS) for the CNMF Problem

```
Input: An $N \times T$ matrix $\tilde{X}$, dimensions $L, K, R = KL$,
    and a parameter $t$.
Output: $\tilde{H}, \tilde{W}_1, \dots, \tilde{W}_L$.
$\tilde{V} \leftarrow$ OrConSPA($\tilde{X}$, $R$, $t$).
$\tilde{G} \leftarrow \operatorname{argmin}_{G \geq 0} \|\tilde{V}G - \tilde{X}\|$.
$\mathcal{C}_1, \dots, \mathcal{C}_k \leftarrow$ ShiftCluster($\tilde{G}$, $K$, $L$).
$\pi_k \leftarrow$ ShiftSort($\tilde{G}$, $L$, $\mathcal{C}_k$) for all $k = 1, \dots, K$.
$\tilde{W}_\ell[:, k] \leftarrow V[:, \pi_k(\ell)]$ for all $k = 1, \dots, K$ and
    $\ell = 1, \dots, L$.
Let $a = \min(L, T - j)$, and define $\tilde{H}$ to be a $K \times T$
    matrix given by
        $$H[k, j] = \frac{1}{a} \sum_{\ell=1}^{a} G[\pi(\ell), j + \ell - 1].$$
return $\tilde{H}, \tilde{W}_1, \dots, \tilde{W}_L$
```

A. Description of the LECS Algorithm

At a high level, our approach will be as follows: we will use the successive projection algorithm (SPA, [24]) to identify the column indices $\mathcal{C}$ of $\tilde{X}$ that correspond to scaled columns of V, then leverage this noisy version of

V to obtain some estimate of G. Finally, we will cluster and sort the rows of G to find the rows of H up to some error. The proposed Locate-Estimate-Cluster-Sort (LECS) algorithm is broken down into four main steps.

• **Locate**. First, LECS 'locates' the column indices $\mathcal{C}$ of $\tilde{X}$ that correspond to $V' = VAD_{VA}$; the columns of VA are present in X, and accordingly their noisy variants are present in $\tilde{X}$, so locating the column indices $\mathcal{C}$ such that $VA = X[:, \mathcal{C}]$ will identify a scaled, permuted variant of V. The algorithm used a modified version of SPA [24], which we call Oracle Conic SPA (OrConSPA); see Algorithm 3 and Appendix A for more details. This method first removes the columns of $\tilde{X}$ that have a low signal to noise ratio using a hyperparameter t. Then, the algorithm rescales the columns of $\tilde{X}$ in order to approximately project the columns of $\tilde{X}$ from the conic hull of the columns of VA onto the convex hull of the columns of VA (the projection is approximate because of the noise). Finally, the algorithm applies SPA to recover the index set $\mathcal{C}$. The name Oracle Conic SPA comes from the 'oracle' used to select the hyperparamter t and from the fact that the algorithm allows X to be in the conic hull of VA, in contrast to standard SPA which requires X to be in the convex hull of VA. This method requires $\mathcal{O}(NTKL)$ operations [24], applying SPA being the most expensive step.

• **Estimate.** In this step, LECS estimates the rows of $G' = D_{VA}^{-1}A^{-1}G$ using nonnegative least squares (NNLS). Estimating G' is essential because the convolutive structure is contained in its rows, which are shifted variants of the rows of H. Solving a convex NNLS problem up to any given precision can be performed in polynomial time using an interior point method (IPM). However, IPMs are second-order methods and hence are computationally demanding. We therefore instead use the block pivot method from [27] which requires one least squares solve per iteration. Each least squares solve takes $\mathcal{O}(NK^2L^2 + NTKL)$ operations, and the algorithm almost always converges after a few iterations.

• **Cluster.** LECS then clusters the rows of G' into K groups $\mathcal{C}_1, \ldots, \mathcal{C}_K$ according to which row of H they are shifted variants of. One cluster should contain the L shifted variants of one row of H. The algorithm achieves this by computing the L-shift similarity between every pair of rows in G', then greedily constructing clusters by adding the available row with the highest average similarity to the rows in the cluster. This simple greedy procedure requires $\mathcal{O}(TK^2L^3)$ operations; see Algorithm 4 and Lemma A.8.

• **Sort.** Finally, within each cluster C_k, LECS sorts the rows of G' based on their shifted similarity to the other rows in the cluster. The algorithm uses a comparison-based sorting algorithm with a comparison operator $\leq$ defined by shifted angle scores. In particular, for two indices i, j, we have $i \leq j$ if $G[j,:]$ can be better expressed as a shifted copy of $G[i,:]$ than the other way around, measured via the cosine between the two vectors. This comparison operator is guaranteed to produce a strict, consistent ordering when the criteria of Theorem 3 are satisfied. However, note that, when these conditions fail to hold there is no guarantee that the comparison operator will produce a coherent ordering. Once this comparison operator has been defined, we can use any comparison-based sorting algorithm (such as merge sort or quick sort) to order the cluster C_k; in our implementation, we use a simple selection sort. When the operator is not consistent (that is, $i \leq j$, $j \leq k$ but $i \nleq k$ for some indices i, j, k), we select at each step the index which is less than or equal to the most other vectors (breaking ties arbitrarily). This sorting procedure requires $\mathcal{O}(TL^3)$ operations; see Algorithm 5 and Lemma A.9.

Finally, using the clustering and sorting, the algorithm reconstructs $W_1, \ldots, W_\ell$ rearranging the columns of V. Each row of H is constructed by taking the de-shifted average of the rows of G in a particular cluster. The total computational cost of Algorithm 1 is $\mathcal{O}(NTKL + TK^2L^3)$ operations plus the time to solve the nonnegative least squares problem.

Algorithm 2 Successive Projection Algorithm ([25], SPA)

```
Input: An $N \times T$ matrix $\tilde{X}$ and a parameter $R \in \mathbf{N}$.
Output: An index set $J$ with $|J| = R$.
  Let $B = \tilde{X}$, $J = \{\}$, $r = 1$.
  while $B \neq 0$ and $r \leq R$ do
      $p \leftarrow \text{argmax}_j \, \|B[:,j]\|_2$.
      $B \leftarrow \left(I - \frac{B_{:p}B_{:p}^T}{\|B_{:p}\|_2^2}\right) B$.
      $J \leftarrow J \cup \{p\}$.
      $r \leftarrow r + 1$.
  end while
  return $J$.
```

Algorithm 3 Oracle Conic SPA (OrConSPA)

```
Input: An $N \times T$ matrix $\tilde{X}$, the parameter $R \in \mathbf{N}$, and
    a threshold value $t > 0$.
Output: An index set $J$ with $|J| = R$.
Define a diagonal matrix $Y$ by
$Y_{jj} = \begin{cases} 0 & \text{if } \|\tilde{X}[:,j]\|_1 \leq t \\ \tilde{X}[:,j] & \text{otherwise} \end{cases}.$
$\tilde{X}' \leftarrow \tilde{X}Y$.
$J \leftarrow$ SPA($\tilde{X}'$, $R$).
return $J$.
```

Algorithm 4 Shift Cluster Algorithm

```
Input: A $KL \times T$ matrix $\tilde{G}$ and parameters $K$, $L$.
Output: $\mathcal{C}_1, \ldots, \mathcal{C}_k$
$\mathcal{A} = \{1, \ldots, KL\}$
for $k = 1, \ldots K$ do
    Choose $i_1 \in \mathcal{A}$ arbitrarily.
    Find $i_2, \ldots, i_L$ by solving
    $\text{argmax}_{i_2,\ldots,i_L \in \mathcal{A}} \sum_{\ell=2}^{L} \cos_L(\tilde{G}[i_1,:], \tilde{G}[i_\ell,:])$.
    $\mathcal{C}_k \leftarrow \{i_1, \ldots, i_L\}$, $\mathcal{A} = \mathcal{A} \setminus \{i_1, \ldots, i_L\}$.
end for
return $\mathcal{C}_1, \ldots, \mathcal{C}_k$.
```

Algorithm 5 Shift Sort Algorithm

```
Input: A $KL \times T$ matrix $\tilde{G}$, the parameter $L$, and a set
    of $L$ indices $\mathcal{C}$.
Output: A map $\pi : \{1, \ldots L\} \to \mathcal{C}$.
For each $i, j \in \mathcal{C}$, compute
    $\mu_{\text{left}}(i,j) = \max_{\ell=0,\ldots,L-1} \cos(S_\ell^T \tilde{G}[i,:] - \tilde{G}[j,:])$,
    $\mu_{\text{right}}(i,j) = \max_{\ell=0,\ldots,L-1} \cos(\tilde{G}[i,:] - S_\ell^T \tilde{G}[j,:])$.
Define a comparison operator on two indices $i, j$ by
    $(i \leq_{L,\tilde{G}} j) = \begin{cases} \text{True} & \text{if } \mu_{\text{left}}(i,j) \geq \mu_{\text{right}}(i,j) \\ \text{False} & \text{otherwise} \end{cases}.$
Sort $\mathcal{C}$ using the comparison operator $\leq_{L,\tilde{G}}$. Let $\pi$ be
    the resulting indexed list (with indices $\{1, \ldots, L\}$).
return $\pi$.
```

B. Guarantee for the Exact Problem

The exact problem is when $E = 0$. Then **[Bounded noise]** is satisfied with $\epsilon = 0$. In this case, Algorithm 1 provably recovers H and each W_ℓ for any $\delta > 0$ and any A such that $\min_i A_{ii} > 0$.

Theorem 2 (Exact Recovery)**.** *Given a CNMF problem that is convolutive separable with respect to $\epsilon = 0$ and some δ, A, Algorithm 1 with inputs $\tilde{X} = X, K, L$ recovers the unique factorization $H, W_1, \ldots, W_\ell$ satisfying (A), (B), and (C), up to a permutation and scaling, in polynomial time; more precisely $\mathcal{O}(NTKL + TK^2L^3)$ flops plus one NNLS solve.*

Proof of Correctness. Because of conditions **[Full Rank]** and **[Separable]**, OrConSPA is guaranteed to find the columns of V (see Lemma A.2). **[Full Rank]** also guarantees nonnegative least squares will return the rows of G (see Lemma A.4). All that remains is to show the grouping correctly recovers W and H.

Given any two rows g_i, g_j that are both shifted versions of some row of H, their similarity is $\cos_L(g_i, g_j) = 1$. Rows that are not shifted variants of one another cannot have similarity greater than or equal to $1 - \delta$ by condition **[Sequentially unique]**. This means the grouping given by the clustering step is necessarily correct.

[Sequentially unique] also entails that the ordering from the sort step is also correct. Since the ordering is correct, the construction of H and each W_ℓ must also be correct up to a permutation and scaling of the rows of H.

Uniqueness follows from the success of the algorithm. Suppose we have two factorizations $W_1, \ldots, W_L, H$ and $W_1', \ldots, W_L', H'$ such that $X = \sum_{\ell=1}^{L} W_\ell H S_{\ell-1} = \sum_{\ell=1}^{L} W_\ell' H' S_{\ell-1}$. Then given X, Algorithm 1 recovers the first factorization up to permutation and scaling. However, Algorithm 1 also recovers the latter factorization up to permutation and scaling. Hence it must be that one factorization is a permute, scaled version of the other, that is, $H = \Pi \Lambda H'$ for some permutation matrix $\Pi \in \{0,1\}^{K \times K}$ and some diagonal matrix $\Lambda \in \mathbf{R}^{K \times K}$ with positive diagonal elements. □

C. Guarantee for the Noisy Problem

Even when $E \neq 0$, we can still recover noisy estimates of $H, W_1, \ldots, W_\ell$ given sufficiently low noise levels.

Theorem 3 (Noisy Recovery)**.** *Suppose a CNMF problem with inputs $\tilde{X}, L, K$ is convolutive separable with respect to δ, ϵ, A and consider some parameter $t > 0$.*

Let $V' = VAD_{VA}$ and $G' = D_{VA}^{-1}A^{-1}G$ and suppose

$$\epsilon + t < \|VA\|_{-1,col}, \tag{2}$$

$$\frac{\epsilon}{t} < C_a\left(\frac{\sigma_{\min}(V')}{\sqrt{KL}\kappa(V')^2}\right), \tag{3}$$

$$\epsilon < C_b\left(\frac{\rho\delta}{\|G'\|_{2,col}\sqrt{T}}\right), \tag{4}$$

where $\rho = C_a\left(\frac{\sigma_{\min}(V')}{\sqrt{KL}\kappa(V')^2}\right) - \frac{\epsilon}{t}$ and C_a, C_b are universal constants independent of all other terms. Furthermore, without loss of generality assume that $\|VA\|_{2,col} > 1$ and $t > 1$ (see below). Then in polynomial time, more precisely $\mathcal{O}(NTKL + TK^2L^3)$ flops plus one NNLS solve, Algorithm 1 finds $\tilde{H} \in \mathbf{R}^{K\times T}$ with bounded error, in the sense that there exists some permutation P such that

$$\min_{1\leq j\leq K} \cos\left(h_j, \tilde{h}_{P(j)}\right) \geq 1 - \frac{\delta}{2},$$

where $h_j, \tilde{h}_{P(j)}$ are the jth and $P(j)$th rows of H and $\tilde{H}$, respectively.

The proof of this theorem is deferred to Appendix C, where we prove a more general statement in Lemma A.10. The assumptions $\|VA\|_{2,col} > 1$ and $t > 1$ are only used to simplify (4) and make the bound easier to read. These assumptions are also made without any loss of generality—we can always consider the equivalent problem with αV and $\alpha\tilde{X} = \alpha X + \alpha E$ for an arbitrarily large $\alpha > 0$; in particular for $\alpha = \max(t^{-1}, \|VA\|_{2,col}^{-1})$. This scaling does not impact the output of the algorithm (beyond scaling), and cancelling terms shows it also does not affect inequality (2) and inequality (3). However, one should note this scaling does impact $\|G'\|_{2,col}$ since G' depends on D_{VA}^{-1}, which changes inequality (4).

Algorithm 1 depends on selecting a good value for the parameter t; the best error bounds is obtained as $t \to \|VA\|_{-1,col} - \epsilon$. This parameter is used to threshold columns of $\tilde{X}$ with a small norm, since these columns will have relatively low signal-to-noise ratios and cause SPA to fail after scaling $\tilde{X}$. In practice, one can run the algorithm several times using different values of t and keep the best fit according to some heuristic (for example, mean square reconstruction error). In fact, since the conditions of the Theorem 3 imply that noisy versions of the columns of VA are present in $\tilde{X}$, we can be certain that $\|\tilde{X}[:,p]\|_{-1,col}$ is within ϵ of $\|VA\|_{-1,col}$ for some $p \in \{1,\ldots,T\}$. So we could run the algorithm T times with parameter $t = \|X[:,p]\| - 2\epsilon$ for $p = 1,2,\ldots,T$ and guarantee that one of the runs yields the desired solution.

IV. Generalizations

One strength of our approach is that any of the subroutines used for the four steps (location, estimation, clustering, and sorting) can be substituted in exchange for different subroutines. For example, we use SPA to locate the columns of V, but could substitute any separable NMF algorithm in its place. In this section, we provide several examples of improvements to LECS obtained by appropriate substitutions and highlight the generality of the high level algorithm.

a) Rank-deficient identification of V: We previously mentioned that **[Full Rank]** is required for two reasons: to guarantee the success of the SPA algorithm and to bound the error when estimating G via nonnegative least squares. However, if we replace SPA with another separable NMF algorithm that does not require V to have full column rank, then the first reason is no longer necessary. SNPA in particular essentially requires the weaker condition that no column of V can be written as a conic combination of the other columns of V. By using this method in place of SPA, we can complete the location step with weaker condition than **[Full Rank]**. This condition is also not necessary for estimating G, which we discuss later in the section.

b) Sufficiently Scattered Condition: Since **[Separable]** is only used in identifying the columns of V, we could also use the *sufficiently scattered* [28], [3], which is more general and allows one to determine V by identifying the minimum-volume NMF of X. Unfortunately, this more general condition suffers from two drawbacks: first, it has yet to be proven robust to noise, and, second, there are currently no algorithms that are guaranteed to find the minimum-volume NMF in polynomial time [3] (this problem is NP-hard in general, but it remains unknown whether this is still true under the sufficiently scattered condition).

c) Rank-deficient Estimation of G: Under some assumptions, the matrix V need not be full rank to properly estimate G. In particular, [26] gives perturbation bounds in the rank-deficient case in a more general setting; these bounds apply to our problem when the rank of $\tilde{V}$ is the same as the rank of V.

d) Robust Least Squares: When we estimate G, we solve the problem $\min_{G\geq 0}\|\tilde{V}G - \tilde{X}\|_F^2$ in an attempt to find some $\tilde{G}$ with rows similar to those of G. If we know the noise level ϵ in advance or have some estimate of it, the optimal estimate is given by the robust optimization problem: $\min_{\tilde{G}\geq 0}\max_{(V,G,X)\in\mathcal{C}}\|\tilde{G} - G\|_{2,row}$, where $\mathcal{C} = \{(V,G,X) : \|V - \tilde{V}\|_{2,col} \leq \epsilon_V, \|X - \tilde{X}\|_{1,col} \leq$

ϵ_X, V, G, X satisfy (1) and (A)-(D)$\}$, where ϵ_V, ϵ_X depend on ϵ. In general, a number of least squares-like procedures may be used in place of standard nonnegative least squares in order to minimize the worst case error $\|\tilde{G} - G\|_{2,row}$.

e) Generalized Similarity Metric: The L-shift similarity measure utilizes the cosine of the angle between two vectors because it is scale invariant, allowing us to consider a and αa equivalent, where $a \in \mathbf{R}^n$ and $\alpha \in \mathbf{R}, \alpha > 0$. The choice of similarity measure heavily influences the clustering and sorting steps, as seen in the proofs of Lemma A.8 and Lemma A.9. Therefore, a wise adjustment of this measure could lead to significant improvements in the final two steps of LECS. This might include a simple post-processing step, such as exponentiation, or an entirely different similarity measure. See [29], [30] for a more comprehensive review.

f) Spectral Clustering: Our greedy clustering algorithm is simple but suboptimal, in the sense that it does not take advantage of the global cluster structure. One more advanced approach is to use a spectral clustering method [29]. Let the matrix M be defined as

$$M_{ij} = \begin{cases} 1 & \text{if } \exists k \in \{1, \dots, K\} \text{ such that } i, j \in \mathcal{C}_k, \\ 0 & \text{otherwise.} \end{cases}$$

Let $v_1, \dots, v_k$ be the eigenvectors corresponding to the K largest eigenvalues of M. Orient each eigenvector so that $|\max_j (v_i)_j| \geq |\min_j (v_i)_j|$ (that is, ensure the entry with the largest magnitude is positive). Then the indices of the largest L entries in v_i form the representatives for cluster i. The challenge is to construct a similarity matrix $\tilde{M}$ close to M using only the observed data. One option is to construct $\tilde{M}$ using the L-shift similarity, then set the largest KL^2 entries to 1 and the remaining entries to 0. Under the conditions of Theorem 3, $\tilde{M} = M$ and therefore recovery is guaranteed. Results from matrix perturbation theory, such as the Davis-Kahan bound [31], suggest there is a strong theoretical justification for using a spectral clustering method, given the right construction of $\tilde{M}$.

V. Numerical Experiments

We first test the LECS algorithm on synthetic data and verify it correctly finds the ground truth in low noise settings. Then, we demonstrate that LECS finds reasonable results on the spectrogram of a songbird and can be used as an effective initialization for other algorithms like multiplicative updates [5] and alternating nonnegative least squares (ANLS) [14].

All code is written in Julia [32] and available on GitHub at github.com/degleris1/CMF.jl. In particular, the code used at the time of publication is available in release v0.1. We use the code from github.com/ahwillia/NonNegLeastSquares.jl to solve NNLS problems via the pivot method. The figures for Sections V-A and V-B are produced by the Jupyter notebooks `figures/sep_synth.ipynb` and `figures/sep_song.ipynb`, and the LECS algorithm itself is available file `src/algs/separable.jl`. Note that this repository is available to other researchers as a tool for fitting CNMF models, as well as rapidly developing and testing new CNMF algorithms.

A. Unstructured Synthetic Data

In this experiment, we consider synthetic data that is separable by construction, but lacks structure in the sense that matrices are generated randomly without constraints on important parameters like condition number. Since angle determines the uniqueness of a vector up to scaling and plays an important role in our theoretical analysis, we use it to measure performance. Specifically, after the algorithm estimates $\tilde{H}$, the score is computed as the average cosine of the angles between the rows of $\tilde{H}$ and the rows of H (minimized over all row permutations). This is formally given by

$$\text{score}(H, \tilde{H}) = \min_{P \in \mathcal{P}} \quad \frac{1}{K} \sum_{k=1}^{K} \cos(H[k,:], \tilde{H}[P(k),:]),$$

where $\mathcal{P}$ is the set of all permutations. We test four different algorithms:

- **LECS.** The LECS algorithm as presented in Algorithm 1.

- **LECS-Pre.** The LECS algorithm with SPA replaced by heuristically preconditioned SPA from [33]. Heuristically preconditioned SPA computes the SVD $U\Sigma V^T$ of $\tilde{X}$, then runs SPA on $\Sigma^{-1} U^T \tilde{X} = V^T$. Under the assumption that the data points are evenly distributed within the convex cone generated by V, it is more robust than SPA [34]. This illustrates the flexibility of LECS with respect to the choice of the different building blocks; see Section IV.

- **Mult.** The multiplicative updates algorithm for minimizing the Frobenius norm used in [5], [7]. This also corresponds to the β-divergence update rule from [11], [12] when $\beta = 2$. The algorithm is initialized randomly and scaled using the method from [35]. The alternating update rules runs for 5 seconds.

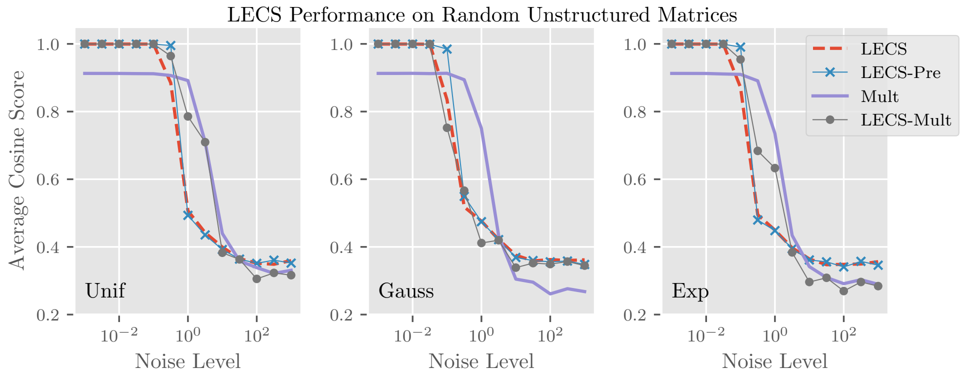


Fig. 1: Performance on unstructured synthetic data.

- **LECS-Mult.** The same multiplicative updates algorithm as before, except using LECS as its initialization.

For each $\ell \in \{1, \dots, L\}$, we let $(W_\ell)_{ij} \sim \mathbf{Unif}(0.5, 1.5)$ for all i, j, where $\mathbf{Unif}(a, b)$ is the uniform distribution in the interval $[a, b]$. We let $H_{ij} \sim \mathbf{Unif}(0, 1)\mathbf{Bern}(1 - p)$ for all i, j, where p is our sparsity parameter and $\mathbf{Bern}(1 - p)$ is the Bernoulli distribution with parameter $1 - p$. To enforce the separability condition, we then set some entries of H to zero as follows. For each row $H[k, :]$ of H, choose two random $t_k, s_k \in \{0, \dots T - L\}$. For all t satisfying either $t_k - L \leq t \leq t_k + \lfloor L/2 \rfloor$ or $s_k - \lfloor L/2 \rfloor \leq t \leq s_k + L$, we set $H[:, t] = 0$. Then we let $H[k, t_k], H[k, s_k] \sim \mathbf{Unif}(0.5, 1.5)$. Finally, we ensure each $t_1, s_1, \dots, t_K, s_K$ are sufficiently far apart from one another so that the same entries are never set to zero twice. This construction ensures the first half of each sequence ($W_{::k}$) is separable at one location, and the second half at another location. We choose "half-sequences" to demonstrate the full sequence need not be separable at the same location. To generate the noise, we vary a noise parameter β and use one of three procedures: for all i, j

- **Uniform.** Set $E_{ij} \sim \mathbf{Unif}(0, \beta)$.
- **Gaussian.** Set $E_{ij} = \max(-X_{ij}, M_{ij})$ where $M_{ij} \sim \mathbf{Normal}(0, \beta)$.
- **Exponential.** Set $E_{ij} \sim \mathbf{Exponential}(\beta)$.

We generate $n_{\text{trials}} = 10$ random matrices X according the above construction with $N = 100$, $T = 250$, $K = 3$, $L = 5$, $p = 0.75$. Then, for each matrix, the noise level β is varied across $n_{\text{noise}} = 13$ different values spaced logarithmically between $\beta = 10^{-3}$ and $\beta = 10^{3}$.

The results of these experiments are displayed in Figure 1. We observe that for all three noise types, LECS finds the true H for small noise levels, as expected by Theorem 3. Additionally, as expected, the preconditioning improves its tolerance to noise. In comparison, Mult fail to find the true H even in the lowest noise settings. In fact, although Mult finds a solution with low reconstruction error, it is not able to recover the ground truth factor H: this is due to the non-uniqueness problem with CNMF (see Section I). As a matter of fact, Mult with LECS as an initialization keeps the true solution and outperforms random initializations. At higher noise levels, the effects are varied. In particular, LECS-Mult exhibits different performance depending on the type of noise, and notably performs the best (relative to random initialization) when the noise is Gaussian. This is likely due to the fact this update rule attempts to minimize the reconstruction error using a Frobenius loss function, which is a natural objective for Gaussian noise but not uniform nor exponential noise. A different loss function is likely necessary to identify the ground truth given these noise models; for example, entrywise ∞-norm and 1-norm losses may be more appropriate for uniform and exponential noise, respectively.

B. *Songbird Spectrogram*

In practice, noise levels are often too high for the LECS algorithm to identify the ground truth, even when the separability assumptions are satisfied. However, LECS can be used as an initialization to other algorithms for CNMF to achieve faster convergence than random initialization, and to obtain better solutions.

In this experiment, we fit a spectrogram of a singing bird from [7]; the authors have kindly made this dataset publicly available at github.com/FeeLab/seqNMF. The spectrogram matrix $\tilde{X}$ has 141 rows (DFT bins) and 4440 columns (timebins). We fit the spectrogram using the LECS algorithm with $K = 3$, $L = 20$, and a threshold value $t = 10$. Then, we use this initialization to run Mult [5] and ANLS [14] algorithms for 15 and 60 iterations, respectively. Note that ANLS, as for NMF [27], solves the subproblem exactly for W and H using an active set method, as opposed to Mult that is a gradient-like descent method. The threshold value t was chosen after sweeping over all values from 0 to 50 (approximately the maximum 1-norm of a column of $\tilde{X}$).

Figure 2 displays the loss curves for Mult and ANLS using both random initialization and the output of LECS. The loss is measured as the relative mean square error, defined as

$$\mathrm{MSE}_{\mathrm{rel}} = \frac{\|\tilde{X} - \sum_{\ell=1}^{L} W_\ell H S_{\ell-1}\|_F}{\|\tilde{X}\|_F},$$

where $W_1, \ldots, W_L, H$ are the estimated matrices.

We observe that when either Mult or ANLS are initialized using the output of LECS, they converge to a local minimum in fewer iterations. This suggests that LECS finds a solution near some local minimum and could be used to accelerate algorithms on very large datasets. In fact, when used as an initialization for ANLS, LECS allows to obtain a high quality solution faster than random initializations. For Mult, using LECS as an initialization does not perform as well because, on average and after sufficiently many iterations, the solution generated by random initializations have lower reconstruction error. We suspect that the reason is that Mult does not deal well with input matrices with many small entries (which LECS generates) because of the zero locking phenomenon–zero entries cannot be modified by Mult while it takes many iterations for small entries to get large; see for example the discussion in [2] and the references therein. In fact, with the LECS initialization, ANLS reduces the relative error to 56.6% after 15 iterations, while MU reduces it to only 58.4% after 60 iterations. Note also that ANLS performs on average better than MU with an average relative error of 56.6% for ANLS vs. 57.7% for MU; this was already observed [14].

Moreover, despite a high relative mean square error, Figure 3 demonstrates that LECS outputs factors and a reconstruction with qualitative similarities to the ground truth. In fact, after just a single iteration of the ANLS algorithm, the factors visibly mimic the sequential structure in the dataset. Note that, as opposed to the experiments on the synthetic data set performed in the previous section, we cannot assess the quality of the generated factors as the ground truth sources are not available.

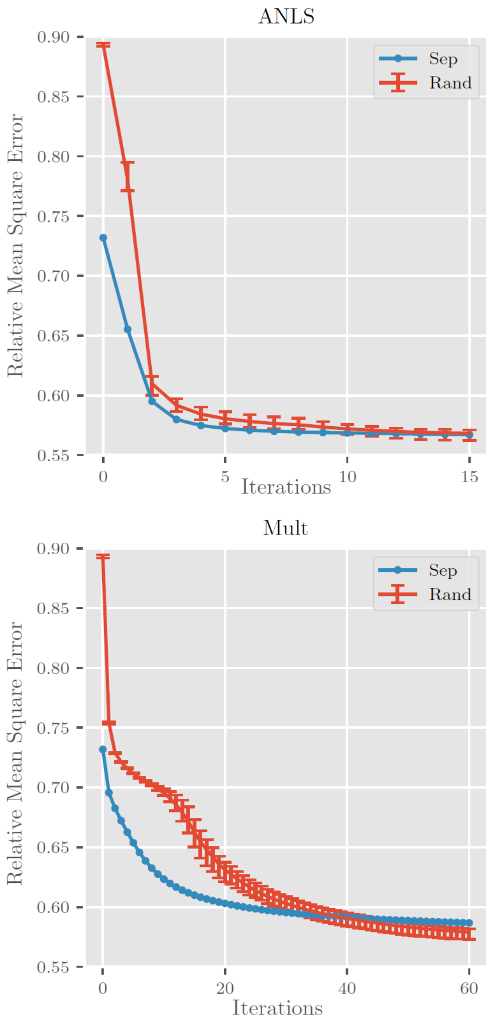
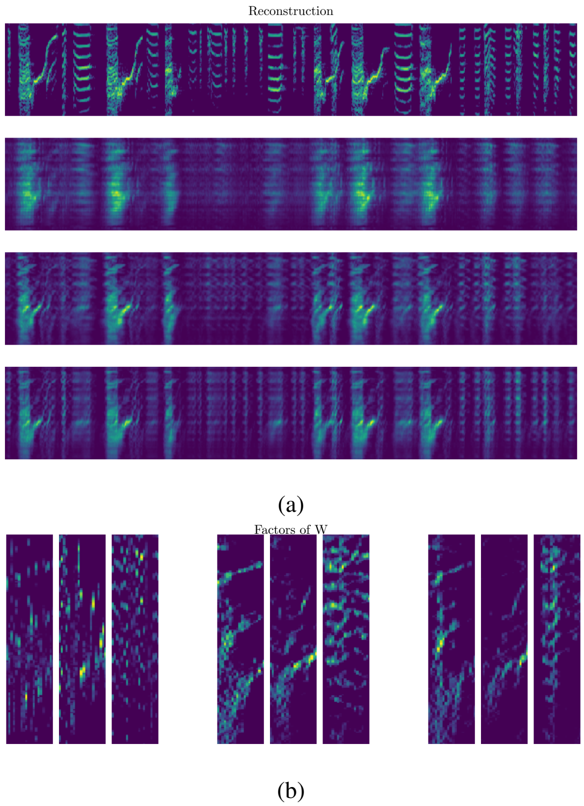


Fig. 2: Loss plots from fitting the songbird spectrogram using both random initialization (median from 30 trials) and the output of LECS as initialization. Error bars represent the 10th and 90th percentiles of the 30 trials.

VI. Conclusion

In this paper we have presented a provably correct and robust algorithm for CNMF under separability conditions. By addressing the questions of identifiability and provable recovery of the solution, this paper offers a novel theoretical perspective on the CNMF problem. Our algorithm draws directly from methods for separable NMF and is easily generalizable—the high level algorithm does not require any particular method for each of its four steps.



(a)

(b)

Fig. 3: Qualitative results on the songbird spectrogram. The top figure **(a)** contains 1000 timebins of the true spectrogram (top), the reconstruction produced by LECS (middle top), and the reconstruction after a single and two iterations of ANLS initialized with LECS (middle bottom and bottom). The bottom figure **(b)** shows the factors $W_{::1}, W_{::2}, W_{::3}$ produced by the LECS algorithm (left three; see Section I for the definition of $W_{::k}$) and after a single iteration and two iterations of ANLS initialized with LECS (middle three and right three).

Future work includes addressing two significant weakness of our approach. First, the location step does not leverage the convolutive structure of CNMF (and could therefore possibly be made more tolerant to noise). Second, the method cannot handle the case when some $W_{::k}$ has repeated columns, which is relevant to processing music datasets that often contain pure harmonic tones.

Appendix

The technical material to follow is divided into three sections. In Appendix A, we discuss a 'conic' variant of SPA which does not require the columns of G to have unit ℓ_1 norm. In Appendix B, we cite or prove results on least squares perturbation, cosine similarity, clustering, and sorting. These results are later used in the proof of our main result, Theorem 3. Finally, we prove Theorem 3 in Appendix C. The full version of the theorem is presented as Lemma A.10, and the simplified Theorem 3 presented in Section III is derived thereafter.

A. *Conic SPA Procedure*

As mentioned in the introduction, SPA is a sequential algorithm for separable NMF that identifies the columns of V among the columns of $X = VG$. At each step, it first extracts the column of X that has the largest ℓ_2 norm, and then project all columns of X onto the orthogonal complement of the extracted column; see Algorithm 2. Under the assumptions that $V = X[:, \mathcal{C}]$ is full column rank and the columns of G have unit ℓ_1 norm, SPA recovers the set V, even in the presence of noise. In this section we generalize SPA from an algorithm for finding the vertices of a polytope (because the columns of G are required to have unit ℓ_1 norm) to an algorithm for finding the extreme rays of a cone. In the noiseless case, it is relatively straightforward to show this generalization by an appropriate rescaling of X. However, when noise is present, extra steps must be taken to ensure points that are primarily noise do not grow too large and are mistaken for a vertex. In fact, in the robustness analysis of SPA, it is assumed from the scratch that the input noisy data matrix has the form $\tilde{X} = VG + E$, where the columns of G are normalized which is key for SPA to succeed. If G is not normalized, then normalization of the input matrix is necessary and this might increase the noise drastically; for example for the columns of X with very small norm. Therefore, to make SPA robust to noise in this scenario, we will need to first remove the columns of $\tilde{X}$ with small norm.

Lemma A.1 (SPA)**.** *[24, Theorem 3] Let $\tilde{X} = X + E = VG + E \in \mathbf{R}^{N\times T}$, where $V \in \mathbf{R}^{N\times R}$ has rank R and $G \in \mathbf{R}_+^{R\times T}$ satisfies $\sum_i G[i,j] \le 1$ for all $1 \le j \le T$. Additionally, suppose that $G = [I \;\; M]\Pi$ for some permutation matrix Π and $M \in \mathbf{R}_+^{R\times T-R}$.*

Let $\|E[:,i]\| \le \epsilon$ for all i with $\epsilon < \frac{C_1\sigma_{\min}(V)}{\sqrt{R}\kappa(V)^2}$, where C_1 is a global constant. Let J be the index set extracted by SPA, which takes time $\mathcal{O}(NTKL)$. Then there exists a permutation P of $\{1,\ldots,R\}$ such that

$$\max_{1\le j\le R} \|\tilde{X}[:,J(j)] - V[:,P(j)]\| = C_2\epsilon\kappa(V)^2,$$

where C_2 is a global constant.

Note that Lemma A.1 makes two strong assumptions on G; first, that $\sum_i G[i,j] \le 1$ for all $1 \le j \le T$, and second, that G contains the identity matrix as a submatrix. Because of the additional structure of the convolutive NMF problem, it is unreasonable to have both these assumptions simultaneously. Instead, we drop the first assumption and generalize the SPA algorithm to any G such that $G = [I\ M]\Pi \in \mathbf{R}_+^{R\times T}$, where Π is a permutation matrix and $M \in \mathbf{R}_+^{R\times T-R}$. Note that this formulation is equivalent to $G = [\Lambda\ M]\Pi$, where Λ is some full rank diagonal matrix, since we can always scale the jth row of G by λ_j and scale the jth column of V by λ_j^{-1} without changing VG. We first demonstrate conditions under which we can recover V if some oracle gives us a threshold value t which is greater than the noise level ϵ but smaller than $\|V\|_{-1} - \epsilon$.

Lemma A.2 (Oracle Conic SPA)**.** *Let $\tilde{X} = X + E = VG + E \in \mathbf{R}_+^{N\times T}$, where $V \in \mathbf{R}_+^{N\times R}$ has rank R and $G = [I\ M]\Pi \in \mathbf{R}_+^{R\times T}$ for some permutation matrix Π and $M \in \mathbf{R}_+^{R\times T-R}$. Denote e_j, v_j the jth column of E and V, respectively. Let $\|e_j\|_1 < \epsilon$ for all j and suppose we know some $t > 0$ such that $t+\epsilon < \|V\|_{-1,col}$. Assume*

$$\frac{\epsilon}{t} < \frac{C_1\sigma_{\min}(VD_V)}{2\sqrt{R}\kappa(VD_V)^2}. \tag{5}$$

Let J be the index set extracted by OrConSPA with inputs $\tilde{X}, R, t$. Then there exists a permutation P of $\{1,\ldots,R\}$ such that

$$\max_{1\le j\le r} \|(\tilde{X}D_{\tilde{X}})[:,J(j)] - (VD_V)[:,P(j)]\|_2 \\ \le 2C_2(\epsilon t^{-1})\kappa(VD_V)^2, \text{ and} \tag{6}$$

$$\max_{1\le j\le r} \|\tilde{X}[:,J(j)] - V[:,P(j)]\|_2 \\ \le (\|V\|_{1,col} + \epsilon)2C_2(\epsilon t^{-1})\kappa(VD_V)^2. \tag{7}$$

Proof. Let v_j be the jth column of V, and allow the same notation for all other matrices. Since the oracle has given us $t \le \|V\|_{-1} - \epsilon_1$ (where $t > 0$), we know that any column j with $\|\tilde{x}_j\|_1 < t$ cannot satisfy $x_j = v_i$ for any i. Therefore, we can define $Y \in \mathbf{R}^{T\times T}$ as a diagonal matrix given by

$$Y_{jj} = \begin{cases} 0 & \text{if } \|\tilde{x}_j\|_1 < t \\ \|\tilde{x}_j\|_1^{-1} & \text{otherwise} \end{cases}.$$

Now define $\tilde{X}' = \tilde{X}Y$. Consider the matrices $V' = VD_V$, $X' = XD_X = V'G'$, and $G' = D_V^{-1}HD_X$. It must be that the columns of G' sum to one, since the columns of X' and V' sum to one. Furthermore, columns g_j of G with only a single non-zero entry will only be scaled by some scalar α, so $G' = [I\ M']\Pi$ for some $M' \in \mathbf{R}_+^{N\times T-R}$ and permutation matrix Π. Therefore, X', V', G' all satisfy the conditions of Lemma A.1; if we wish to apply SPA to identify V' from $\tilde{X}' = X' + E'$, all that remains is to bound the noise $E' = \tilde{X}' - X'$. To bound the noise, we note that for any non-zero column of $\tilde{X}'$, we have $e_j' = \tilde{x}_j' - x_j' = \frac{x_j+e_j}{\|x_j+e_j\|_1} - \frac{x_j}{\|x_j\|_1} = \frac{e_j}{\|x_j+e_j\|_1} - \left(1 - \frac{\|x_j\|_1}{\|x_j+e_j\|_1}\right)\frac{x_j}{\|x_j\|_1}$, and therefore we have the following bound for $\epsilon' = \|e_j'\|_2$:

$$\begin{aligned}
\epsilon' &= \left\| \frac{e_j}{\|x_j + e_j\|_1} - \left(1 - \frac{\|x_j\|_1}{\|x_j + e_j\|_1}\right)\frac{x_j}{\|x_j\|_1} \right\|_2 \\
&\le \frac{\|e_j\|_2}{\|x_j + e_j\|_1} + \left(1 - \frac{\|x_j\|_1}{\|x_j + e_j\|_1}\right)\frac{\|x_j\|_2}{\|x_j\|_1} \\
&\le \frac{\|e_j\|_2}{\|x_j + e_j\|_1} + \left(\frac{\|x_j\|_1 + \|e_j\|_1 - \|x_j\|_1}{\|x_j + e_j\|_1}\right)\frac{\|x_j\|_2}{\|x_j\|_1} \\
&\le \frac{\|e_j\|_2}{\|x_j + e_j\|_1} + \frac{\|e_j\|_1}{\|x_j + e_j\|_1} \le \frac{2\epsilon}{t}.
\end{aligned}$$

Concerning the zero columns, they cannot be noisy versions of a column of V and therefore setting them to zero is irrelevant to the performance of SPA. Therefore, if we apply SPA to X', then Lemma A.1 and (5) tells us our assumptions are satisfactory to recover a noisy estimate of V' with error $C_2(2\epsilon t^{-1})\kappa(VD_V)^2$. Scaling each column by $D_{\tilde{X}}^{-1}$ gives the error in (7). □

Tighter bounds are achieved as t approaches $\|V\|_{-1} - \epsilon$. In real applications, this oracle is not available and t is unknown. However, we do know that if such a t exists, then $0 \le t \le \|\tilde{X}\|_1$ and that $t = \|X_{:j}\|$ for some j. Since SPA is very fast (essentially $2NTR$ flops), it is reasonable to run OrConSPA for multiple values of t and be certain that one of the outputs locates the columns of V (that is, if such a value of t exists at all). In practice, a measure like mean square error can be used to select the "best" value of t. The previous proof specifically uses SPA to the thresholded and scaled matrix $\tilde{X}' = \tilde{X}Y$. However, there is nothing special about our choice of separable NMF algorithm. In fact, other algorithms such as SNPA [25] or AnchorWords [20] will yield more robust noise tolerances. For a more complete list of separable NMF algorithms, see [19]. In this vein, we can generalize Lemma A.2 to an arbitrary separable NMF algorithm.

Lemma A.3 (Oracle Conic Location)**.** *Let $\tilde{X} = X + E = VG + E \in \mathbf{R}_+^{N\times T}$, where $V \in \mathbf{R}_+^{N\times R}$ and $G = [I\ M]\Pi \in \mathbf{R}_+^{R\times T}$ for some permutation matrix Π and $M \in \mathbf{R}_+^{R\times T-R}$.*

Consider some arbitrary separable NMF algorithm `ALG` *with a maximum input noise tolerance $\epsilon_{in} < m_1(V)$ and a maximum output noise $m_2(V, \epsilon_{in})$. Furthermore, suppose VD_V satisfies any other conditions of* `ALG` *(e.g.*

a condition on the rank). Denote e_j, v_j the jth column of E and V, respectively. Now suppose $\|e_j\|_1 \leq \epsilon$ for all j and suppose we know of some $t > 0$ such that $t + \epsilon < \|V\|_{-1}$ and $\frac{\epsilon}{t} < \frac{1}{2}m_1(VD_V)$. Let J be the index set extracted by OCL with inputs $\tilde{X}, R, t$, and `ALG`. *Then there exists a permutation P of $\{1, \ldots, R\}$ such that $\max_j \|\tilde{X}[:, J(j)] - V[:, P(j)]\|_2 \leq (\|V\|_1 + \epsilon)m_2(VD_V, 2\epsilon t^{-1})$.*

B. Relevant Lemmas

1) Perturbation bounds for least squares:

Lemma A.4 (Least Squares Perturbation, [26] Theorem 3, Remark 2)**.** *Consider full rank matrices $A, \tilde{A} \in \mathbf{R}^{T \times R}$ and vectors $b, \tilde{b} \in \mathbf{R}^T$. Define $F = A - \tilde{A}$ and $e = b - \tilde{b}$ Suppose there is a convex set $\mathcal{C}$ and a vector $y \in \mathcal{C}$ such that $Ay = b$ and let $\tilde{y} = \operatorname{argmin}_{y' \in \mathcal{C}} \|\tilde{A}y' - \tilde{b}\|_2^2$. If $\sigma_{\min}(A) > \sigma_{\max}(F)$, then*

$$\|y - \tilde{y}\|_2 \leq \frac{\sigma_{\max}(F)\|y\|_2 + \|e\|_2}{\sigma_{\min}(A) - \sigma_{\max}(F)}.$$

Corollary A.5. *Suppose we are given $\tilde{X}, X \in \mathbf{R}^{T \times N}$ such that $E = X - \tilde{X}$ satisfies $\|E[:, j]\| \leq \epsilon$ for all j. Additionally, suppose we also have $\tilde{V}, V \in \mathbf{R}^{N \times KL}$ such that $F = V - \tilde{V}$ satisfies $\sigma_{\min}(V) > \sigma_{\max}(F)$. Suppose there is a $G \geq 0$ such that $VG = X$ and let*

$$\tilde{G} = \operatorname*{argmin}_{G' \geq 0} \|\tilde{V}G' - \tilde{X}\|_F^2.$$

Also let $\gamma = \|G[:, j]\|_{2,col}$ be the maximum norm of any column of G. Then the distance between rows of G^* and $\hat{G}$ is bounded by $\max_i \ \|G[i, :] - \hat{G}[i, :]\|_2 \leq \sqrt{T}\left(\frac{\sigma_{\max}(F)\gamma + \epsilon}{\sigma_{\min}(V) - \sigma_{\max}(F)}\right)$.

Proof. For the jth column g_j of G, we have the bound

$$\|g_j - \tilde{g}_j\|_2^2 \leq \left(\frac{\sigma_{\max}(F)\|g_j\|_2 + \epsilon}{\sigma_{\min}(V) - \sigma_{\max}(F)}\right)^2.$$

Then for a row $G[i, :]$ we must have

$$\|G[i, :] - \tilde{G}[i, :]\|_2^2 \leq T\left(\frac{\sigma_{\max}(F)\gamma + \epsilon}{\sigma_{\min}(V) - \sigma_{\max}(F)}\right)^2.$$

Taking the square root yields the desired bound. □

2) Cosine Similarity:

Lemma A.6. *Consider the vectors $y_1, y_2, e_1, e_2 \in \mathbf{R}^T$. Let $\theta = \cos(y_1, y_2) > 0$. Suppose $\|e_1\|_1 \leq \epsilon\|y_1\|_1$ and $\|e_2\|_2 \leq \epsilon\|y_2\|_2$ for $0 < \epsilon < 1$. Then we have*

$$\frac{\theta - 3\epsilon}{(1+\epsilon)^2} \leq \cos(y_1 + e_1, y_2 + e_2) \leq \frac{\theta + 3\epsilon}{(1-\epsilon)^2}.$$

Proof. First consider when $\|y_1\|_2 = \|y_2\|_2 = 1$. We have $\frac{(y_1+e_1)^T(y_2+b_2)}{\|y_1+e_1\|_2\|y_2+b_2\|_2} = \frac{\theta + y_1^T e_2 + y_2^T e_1 + e_1^T e_2}{\|y_1+e_1\|_2\|y_2+e_2\|_2} \geq \frac{\theta - 2\epsilon - \epsilon^2}{\|y_1+e_1\|_2\|y_2+e_2\|_2} \geq \frac{\theta - 3\epsilon}{(1+\epsilon)^2}$. Now consider when $\|y_1\|_2, \|y_2\|_2 \neq 1$. Since cos is scale invariant, we scale the vectors to $(y_1 + e_1)/\|y_1\|_2$ and $(y_2 + e_2)/\|y_2\|_2$ and apply the above statement to obtain the lower bound in our lemma. The upper bound follows similarly. □

Corollary A.7. *Under the same assumptions as in Lemma A.6 and $\epsilon < \frac{1}{2}$, we have*

$$\theta - 35\epsilon \leq \cos(y_1 + e_1, y_2 + e_2) \leq \theta + 35\epsilon.$$

Proof. For the lower bound, use the Taylor expansion of $(1+\epsilon)^{-1}$ to show that

$$\begin{aligned} \frac{\theta - 3\epsilon}{(1+\epsilon)^2} &= (\theta - 3\epsilon)\left[1 - \epsilon + \epsilon^2 - \epsilon^3 + \ldots\right]^2 \\ &\geq (\theta - 3\epsilon)(1-\epsilon)^2 \text{ (since } \epsilon^{2n} - \epsilon^{2n+1} > 0) \\ &\geq (\theta - 3\epsilon)(1 - 2\epsilon) \geq \theta - 5\epsilon. \end{aligned}$$

For the upper bound, consider the Taylor expansion of $(1-\epsilon)^{-1} = \left[1 + \epsilon + \epsilon^2 + \ldots\right] = [1 + \epsilon(1 + \epsilon + \ldots)] = \left[1 + \epsilon\frac{1}{1-\epsilon}\right] \geq (1 + 2\epsilon)$, hence $\frac{\theta+3\epsilon}{(1-\epsilon)^2} \leq (\theta + 3\epsilon)(1 + 2\epsilon)^2 \leq (\theta + 3\epsilon)(1 + 8\epsilon) \leq \theta + 35\epsilon$. □

3) Clustering and Sorting:

Lemma A.8 (Clustering)**.** *Suppose $G \in \mathbf{R}^{KL \times T}$ is produced from (1) for a CNMF problem with some H that satisfies [δ-sequentially unique]. Consider $\tilde{G}$ such that $\max_{1 \leq j \leq KL} \ \|g_j - \tilde{g}_{P(j)}\|_2 = \epsilon < \frac{\delta}{70}\|G\|_{-2,row}$, for some permutation P, where $g_j = G[j, :]$ and $\tilde{g}_j = \tilde{G}[j, :]$. Then in $O(TK^2L^3)$ time we can recover K sets $\mathcal{C}_k$ of L indices such that for each row h_k of H, there is some group $\mathcal{C}_k$ where for all $\ell = 0, \ldots, L-1$, there exists some $j \in \mathcal{C}_k$ such that $g_{P^{-1}(j)} = S_\ell^T h_k$.*

Proof. Fix some k. Consider i, j such that $g_i = S_{\tau_1}^T h_k$ and $g_j = S_{\tau_2} h_k$ for $\tau_1, \tau_2 \in \{0, \ldots L-1\}$. Then we must have one of $g_i = S_\ell g_j$ or $g_j = S_\ell g_i$ for some $\ell \in \{0, \ldots, L-1\}$. Without loss of generality, assume $g_i = S_\ell g_j$. Let $r_i = \tilde{g}_{P(i)} - g_i$ and $r_j = S_\ell^T \tilde{g}_{P(j)} - S_\ell^T g_j$. By assumption, $\max(\|r_j\|, \|r_i\|) < (\delta/70)\|G\|_{-2,row}$. Now we apply Corollary A.7 to $\tilde{g}_i = g_i + r_i$ and $S_\ell^T \tilde{g}_j = S_\ell^T g_j + r_j$ and consider that $\cos(g_i, S_\ell^T g_j) = 1$ to obtain $\cos(S_\ell^T \tilde{g}_j, \tilde{g}_i) > 1 - \frac{\delta}{2}$, and hence $\cos_L(\tilde{g}_j, \tilde{g}_i) \geq 1 - \frac{\delta}{2}$.

Now consider i, j such that $g_i = S_{\tau_1}^T h_k$ for some $\tau_1 \in \{0, \ldots, L-1\}$, but $g_j \neq S_{\tau_2}^T h_k$ for any $\tau_2 \in \{0, \ldots, L-1\}$. Then by [$\delta$-sequentially unique], we must have that $\max(\cos(g_i, S_\ell^T g_j), \cos(S_\ell^T g_i, g_j)) \leq 1 - \delta$ for any $\ell \in$

$\{0, \ldots, L-1\}$. Applying Corollary A.7 with the same construction as before, we have

$$\max(\cos(S_\ell^T \tilde{g}_j, \tilde{g}_i), \cos(\tilde{g}_j, S_\ell^T \tilde{g}_i)) < 1 - \frac{\delta}{2},$$

for any $\ell \in \{0, \ldots, L-1\}$. Therefore, $\cos_L(\tilde{g}_1, \tilde{g}_2) < 1 - \frac{\delta}{2}$. Together, these two angle bounds gives a strict criterion for clustering the rows of G into K groups of L vectors, according to their corresponding row of H. The determine the runtime, we note that we must compute the angle between KL vectors L times each. Computing the angle of each vector requires $\mathcal{O}(T)$ flops, so in total we require $\mathcal{O}(TK^2L^3)$ operations. The grouping of the vectors can be done simply in $\mathcal{O}(K^2L)$ and is comparatively negligible as $K < T$. □

Lemma A.9 (Sorting)**.** *Suppose $G \in \mathbf{R}^{KL \times T}$ is produced from (1) for a CNMF problem with some H that satisfies [δ-sequentially unique]. Suppose there exists a permutation P and a matrix $\tilde{G}$ such that*

$$\max_{1 \le j \le KL} \|g_j - \tilde{g}_{P(j)}\|_2 = \epsilon < \frac{\delta}{70}\|G\|_{-2,row},$$

where $g_j = G[j, :]$, $\tilde{g}_j = \tilde{G}[j, :]$.

Now suppose we have some set of L indices $\mathcal{C}_k$, and that for all $\ell = 0, \ldots, L-1$ there exists $j \in \mathcal{C}_k$ such that $g_{P^{-1}(j)} = S_\ell^T h_k$.

Then in $\mathcal{O}(TL^3)$ time we can recover a bijective map $\pi : \mathcal{C}_k \to \{0, \ldots, L-1\}$ such that $g_{P^{-1}(j)} = S_{\pi(j)}^T h_k$.

Proof. For any $i, j \in \mathcal{C}_k$, a nearly identical argument to Lemma A.8 gives us a decision criterion for finding a unique $\ell_{ij} \in \{0, \ldots, L-1\}$ such that either $g_{P^{-1}(j)} = S_{\ell_{ij}}^T g_i$ or $g_{P^{-1}(i)} = S_{\ell_{ij}}^T g_j$. We find such ℓ for each pair of indices, and construct a mapping $y(i,j) : \mathcal{C}_k \times \mathcal{C}_k \to \{-L+1, \ldots, L-1\}$ defined by

$$y(i,j) = \begin{cases} \ell_{ij} & \text{if } g_{P^{-1}(i)} = S_{\ell_{ij}}^T g_j \\ -\ell_{ij} & \text{if } g_{P^{-1}(j)} = S_{\ell_{ij}}^T g_i \end{cases}.$$

By the construction of G and $\mathcal{C}_k$, there is only one i such that $y(i,j)$ will be strictly nonnegative for all $j \in \mathcal{C}_k$. Then our permutation is π is given by $\pi(j) = y(i,j)$.

Each comparison requires L distance computations, and each distance computations takes $\mathcal{O}(T)$ flops. Since we must measure the distance between L different vectors, this totals to $\mathcal{O}(TL^3)$ flops. The construction of y and π requires $\mathcal{O}(L^2)$ flops and is comparatively negligible. □

C. Proof of the Recovery Guarantee in the Presence of Noise

Lemma A.10 proves the recovery guarantee in slightly more details than in Theorem 3. It can then be used to obtain Theorem 3.

Lemma A.10. *Suppose a CNMF problem with inputs $\tilde{X}, L, K$ is convolutive separable with respect to δ, ϵ, A. Let $V' = VAD_{VA}$ and $G' = D_{VA}^{-1}A^{-1}G$ and suppose we know some $t > 0$ such that*

$$\epsilon + t < \|VA\|_{-1,col}, \tag{8}$$

$$\frac{\epsilon}{t} < \frac{\sigma_{\min}(V')}{2\sqrt{KL}\kappa(V')^2} \min\left(C_1, C_2^{-1}\right) \tag{9}$$

$$\delta > \frac{70\sqrt{T}}{\|G'\|_{-2,row}} \left(\frac{c\|G'\|_{2,col} + \epsilon}{\sigma_{\min}(V') - c}\right), \tag{10}$$

where C_1 and C_2 are universal constants independent of all other terms, V is defined in (1), and $c = 2C_2(\epsilon t^{-1})\kappa(V')^2\sqrt{KL}$. Then in $\mathcal{O}(NTKL + TK^2L^3)$ time and the time for one NNLS solve, Algorithm 1 finds $\tilde{H} \in \mathbf{R}^{T \times K}$ with bounded error, in the sense that there exists some permutation P such that

$$\min_{1 \le j \le K} \cos\left(h_j, \tilde{h}_{P(j)}\right) \ge 1 - \frac{\delta}{2}, \tag{11}$$

where $h_j, \hat{h}_{P(j)}$ are the jth and $P(j)$th rows of H and $\hat{H}$, respectively.

Proof. We break the proof up into several steps.

a) Location: We know that $V = X[:, \mathcal{C}]A\Pi$ for some index set $\mathcal{C}$, some diagonal scaling matrix A, and some permutation matrix Π. Instead of identifying V directly, we identify $VA = X[:, \mathcal{C}]\Pi$, which satisfies $X = VA[I \ M]\Pi'$ for some $M \in \mathbf{R}^{N \times T-R}$ and some permutation matrix Π'. In particular, we apply OrConSPA to inputs $\tilde{X}, R, t$ to recover $V' = VAD_{VA}$. Since $X = VA[I \ M]\Pi'$ and the noise is bounded in (8) and (9) by $\epsilon + t \le \|VA\|_{-1}$ and $\frac{\epsilon}{t} \le \frac{C_1\sigma_{\min}(V')}{2\sqrt{KL}\kappa(V')^2}$, we can apply Lemma A.2 and show that the index set J output by OrConSPA satisfies

$$\max_j \|(\tilde{X}D_{\tilde{X}})[:, J(j)] - V'[:, P(j)]\|_2^2 \le 2C_2\frac{\epsilon}{t}\kappa(V')^2.$$

Let $\tilde{V} = (\tilde{X}D_{\tilde{X}})[:, J(j)]$.

b) Perturbation: Given $\tilde{V}$, we estimate $G' = D_{VA}^{-1}A^{-1}G$ using nonnegative least squares. Let $F = \tilde{V} - V'$. We know the largest singular value is bounded above by the Frobenius norm, so we have $\sigma_{\max}(F) \le \sqrt{KL}\left(2C_2(\epsilon t^{-1})\kappa(V')^2\right) = c$. Rearranging (9) using the latter term of the minimum gives us the bound

$\sigma_{\min}(V') > 2C_2(\epsilon t^{-1})\kappa(V')^2\sqrt{KL} = c$. This allows us to apply Corollary A.5, so the recovered matrix $\tilde{G}$ must satisfy $\max_i \|G'[i,:] - \tilde{G}[i,:]\| \leq \sqrt{T}\left(\frac{\sigma_{\max}(F)\|G'\|_{2,col}+\epsilon}{\sigma_{\min}(V')-\sigma_{\max}(F)}\right)$. We then have

$$\max_{1\leq i\leq R} \|G'[i,:] - \tilde{G}[i,:]\| \leq \sqrt{T}\left(\frac{c\|G'\|_{2,col}+\epsilon}{\sigma_{\min}(V')-c}\right). \tag{12}$$

c) Clustering and Sorting: Since $G' = D_{VA}^{-1}A^{-1}G$, the angle to a row of G' is the same as an angle to a row of G. Rearranging (10) we have

$$\frac{\delta\|G'\|_{-2,row}}{70} > \sqrt{T}\left(\frac{c\|G'\|_{2,col}+\epsilon}{\bar{\sigma}_{\min}(VA)-c}\right).$$

Combining this with (12) lets us apply Lemma A.8 and Lemma A.9 to achieve the correct clustering and sorting. The bound (11) follows automatically. □

Proof of Theorem 3. All we must do is simplify (9) and (10) to match the bounds in Theorem 3. For (9), simply set $C_a = \frac{\min(C_1, C_2^{-1})}{2}$. For (10), we begin by recalling our assumption that $\|VA\|_{2,col} > 1$ and that $t > 1$. We must have that $\|A^{-1}G\|_{-2,row} > 1$ since each row has an entry with a 1 in it by construction. Then since $\|VA\|_{2,col} > 1$ and equivalently $(D_{VA})_{ii}^{-1} > 1$, then $\|G'\|_{-2,row} > 1$. Given these assumptions, we have the sufficient condition

$$\delta > \frac{70\sqrt{T}(c\|G\|_{2,col}+\epsilon)}{\sigma_{\min}(V')-c}.$$

Dividing the numerator and denominator by $2C_2\kappa(V')^2\sqrt{KL}$ leads to $\delta > \frac{70\sqrt{T}(\|G'\|_{2,col}(\epsilon t^{-1})+\epsilon)}{\frac{\sigma_{\min}(V')}{2C_2\kappa(V')^2\sqrt{KL}}-(\epsilon t^{-1})}$. Then we have the sufficient condition

$$\delta > \frac{70\sqrt{T}(\|G'\|_{2,col}(\epsilon t^{-1})+\epsilon))}{\rho},$$

where $\rho = C_a\frac{\sigma_{\min}(V')}{\kappa(V')^2\sqrt{KL}} - (\epsilon t^{-1})$. Since we assumed $t > 1$, $\delta > \frac{140\sqrt{T}\|G'\|_{2,col}\epsilon}{\rho}$ is also sufficient. Setting $C_b^{-1} = 140$ gives us $\delta > C_b^{-1}\sqrt{T}\|G'\|_{2,col}\epsilon/\rho$ which in turn reduces to $\epsilon < C_b\delta\rho/\|G'\|_{2,col}\sqrt{T}$. □

Acknowledgment

The authors thank the reviewers for their feedback which helped improve the paper significantly.

References

[1] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[2] N. Gillis, "The why and how of nonnegative matrix factorization," in *Regularization, Optimization, Kernels, and Support Vector Machines*, J. Suykens, M. Signoretto, and A. Argyriou, Eds. Boca Raton, Florida: Chapman & Hall/CRC, 2014, ch. 12, pp. 257–291.

[3] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 36, pp. 59–80, 2019.

[4] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2004, pp. 494–499.

[5] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-d deconvolution for blind single channel source separation," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2006, pp. 700–707.

[6] J. Zhou, R. Liang, L. Zhao, L. Tao, and C. Zou, "Unsupervised learning of phonemes of whispered speech in a noisy environment based on convolutive non-negative matrix factorization," *Information Sciences*, vol. 257, pp. 115–126, 2014.

[7] E. L. Mackevicius, A. H. Bahle, A. H. Williams, S. Gu, N. I. Denisenko, M. S. Goldman, and M. S. Fee, "Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience," *eLife*, vol. 8, p. e38471, 2019.

[8] S. A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2010.

[9] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[10] W. Wang, A. Cichocki, and J. A. Chambers, "A multiplicative algorithm for convolutive non-negative matrix factorization based on squared euclidean distance," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2858–2864, 2009.

[11] S. Gorlow, A. T. Hariraman *et al.*, "Multiplicative updates for convolutional NMF under β-divergence," *arXiv preprint arXiv:1803.05159*, 2018.

[12] D. Fagot, H. Wendt, C. Févotte, and P. Smaragdis, "Majorization-minimization algorithms for convolutive NMF with the beta-divergence," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8202–8206.

[13] A. H. Phan, A. Cichocki, P. Tichavskỳ, and Z. Koldovskỳ, "On connection between the convolutive and ordinary nonnegative matrix factorizations," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 288–296.

[14] A. Degleris, B. Antin, and A. Williams, personal communication, 2019.

[15] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[16] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization–provably," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 145–162.

[17] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Advances in neural information processing systems*, 2004, pp. 1141–1148.

[18] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote

sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 67–81, 2014.

[19] N. Gillis, "Separable simplex-structured matrix factorization: Robustness of combinatorial approaches," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5521–5525.

[20] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu, "A practical algorithm for topic modeling with provable guarantees," in *International Conference on Machine Learning*, 2013, pp. 280–288.

[21] M. Panov, K. Slavnov, and R. Ushakov, "Consistent estimation of mixed memberships with successive projections," in *International Conference on Complex Networks and their Applications*. Springer, 2017, pp. 53–64.

[22] R. Luce, P. Hildebrandt, U. Kuhlmann, and J. Liesen, "Using separable nonnegative matrix factorization techniques for the analysis of time-resolved Raman spectra," *Applied spectroscopy*, vol. 70, no. 9, pp. 1464–1475, 2016.

[23] U. Araújo, B. Saldanha, R. Galvão, T. Yoneyama, H. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 57, no. 2, pp. 65–73, 2001.

[24] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithmsfor separable nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 698–714, 2014.

[25] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1420–1450, 2014.

[26] P. Lötstedt, "Perturbation bounds for the linear least squares problem subject to linear inequality constraints," *BIT Numerical Mathematics*, vol. 23, no. 4, pp. 500–519, 1983.

[27] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[28] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 211–224, 2013.

[29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[30] P. Favati, G. Lotti, O. Menchi, and F. Romani, "Construction of the similarity matrix for the spectral clustering method: numerical experiments," *arXiv preprint arXiv:1904.11352*, 2019.

[31] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. iii," *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 1–46, 1970.

[32] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[33] N. Gillis and S. A. Vavasis, "Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 677–698, 2015.

[34] N. Gillis and W.-K. Ma, "Enhancing pure-pixel identification performance via preconditioning," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1161–1186, 2015.

[35] N.-D. Ho, "Nonnegative matrix factorization algorithms and applications," Ph.D. dissertation, PhD thesis, Université catholique de Louvain, 2008.