Greedy Conical Hull Algorithms for Separable Nonnegative Matrix Factorization and their Application in Music Source Separation

Project Report
submitted by

Vignesh Sairaj
COE14B042

in fulfilment for the award of the degree of

Bachelor of Technology
in
Computer Engineering

Guide
Dr. B. Sivaselvan
IIITDM Kancheepuram

Indian Institute of Information Technology
Design and Manufacturing, Kancheepuram, India

May 2018

BONAFIDE CERTIFICATE

This is to certify that the thesis titled **"Greedy Conical Hull Algorithms for Separable NMF and Their Application in Music Source Separation"** submitted by **Vignesh Sairaj** to the Indian Institute of Information Technology Design and Manufacturing, Kancheepuram, for the award of Bachelor of Technology in Computer Engineering, is a *bona fide* record of the project work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Sivaselvan B
Project Guide
Assistant Professor
Indian Institute of Information Technology
Design and Manufacturing, Kancheepuram
Chennai - 600 127
India

Place: Chennai
Date: 30th May 2018

ACKNOWLEDGMENTS

I express my sincere thanks to my project supervisor, Dr. B. Sivaselvan, for his constant support and valuable suggestions. I would like to thank him for his patience and guidance throughout the time spent on my project.

Furthermore, I would like to thank the Department of Computer Engineering and all the Faculty of IIITDM who have helped develop an understanding the rudiments of Computer Science & Engineering and an appreciation for its applications in various domains. I would like to thank my professors for their patience and understanding throughout my stay at IIITDM and their support in times of need.

I would also like to thank the review committee for their valuable advice and suggestions which have helped towards the betterment of this project.

Vignesh Sairaj
COE14B042
IIITDM

Contents

List of Figures

Abstract

Audio samples are typically complex mixtures of signals from different sources. Our objective is to decompose this complex sound mixture into its constituent components. This is referred to as **source separation**. In the case of music signals, music-specific properties and additional musical knowledge can be exploited. The number of spectral signatures in music signals is usually fewer than other types of audio and lends itself effectively to **Nonnegative Matrix Factorization (NMF)** techniques.

Given $M_{m\times n}$, element-wise nonnegative, and target rank r, nonnegative matrix factorization is the problem of finding $W_{m\times r}$ and $H_{r\times n}$ (each element-wise nonnegative), such that $M \approx WH$. A matrix $M_{m\times n}$ is said to be **near-r-Separable** if it is "close" to a Matrix $\hat{M}$ that permits a nonnegative factorization of the form $\hat{M} = \hat{M}(:,\kappa)H$, where κ is an r-subset of the columns of $\hat{M}$ and $H_{r\times n} \geq 0$.

In this work, we propose the application of greedy variants of existing conical hull algorithms for the near-separable NMF problem to music audio source separation by assuming a separability condition on the music signal.

Chapter 1

Motivation

1.1 Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) is a linear dimensionality reduction technique to approximate a matrix $M_{m \times n}$ as a product of matrices of rank at most r, where $r << min(m, n)$. Given $M_{m \times n}$ and target rank r, matrix factorization is the problem of finding $W_{m \times r}$ and $H_{r \times n}$, such that $M = WH$. Expressed another way, this is equivalent to taking weighted linear combinations of the columns of W with the corresponding columns of H as the weights for each column in the data matrix M. i.e., $M_{:j} = \sum_{k=1}^{r} W_{:r} H_{kj}$, where $M_{:j}$ denotes the j^{th} column of M. When the input matrix is guaranteed to be nonnegative, and the matrices W and H are also required to be nonnegative, the problem becomes nonnegative matrix factorization. NMF was first introduced by Paatero and Tapper[15] as positive matrix factorization and subsequently popularized by Lee and Seung [13]. NMF has many practical applications where it is preferred over other LDRs due to its purely additive constraints such as in hyper-spectral imaging, text mining, and facial feature extraction. In these application, each column of the data matrix is expressed a linear mixture of features that are combined in a purely additive manner that conforms to inherent model. The decision version of the NMF problem (i.e., determining if a given matrix M supports an exact nonnegative r-factorization) was shown to be NP-Hard by [Vavasis, 2009] [17].

1.2 Music Audio Source Separation

Audio samples are typically complex mixtures of signals from different sources. Our objective is to decompose this complex sound mixture into its constituent components. This is referred to as source separation. The sources of interest could be the voice of a specific speaker from a mixture of conversations with multiple speakers and background noises. This is the well-known *cocktail problem.* The sources could also be different instruments playing together, or a speaker talking in the foreground with music being played in the background, etc. In the case of music signals, music-specific properties and additional musical knowledge can be exploited. The number of spectral signatures in music signals is usually fewer than other types of audio and lends itself effectively to NMF techniques. The matrix M is obtained by performing a Short-Time Fourier Transform (STFT) on the audio signal and mapping the intensities of each bin in the resulting spectrogram into a component of a spectrum-vector. The spectrum vectors of each time interval are grouped together as column vectors to give M. W can be interpreted as the spectral signatures of a few fundamental notes/clusters of notes and H contains the temporal information for when each of these notes is “sounded”. Constrained NMF techniques have been investigated for this purpose in [11], [18], and [16]. We look to investigate the application of near-separable NMF algorithms for the same application.

1.3 Separable NMF

A matrix $M_{m\times n}$ is said to be r-Separable if it permits a nonnegative factorization of the form $M = W[I_r, H']\Pi$, where I_r is the r-Identity matrix and Π an $n \times n$ permutation matrix, and H' is $r \times (n-r)$. This is equivalent to $M = M(:, \kappa)H$, where κ is an r-subset of the columns of M and $H_{r\times n} > 0$. A matrix M is said to be near-separable if it differs from a separable matrix by a noise matrix η for $\|\eta\| < \epsilon$, for some choice of norm $\|\cdot\|$and ϵ. When the columns are looked at as m-dimensional vectors, this is the same as assuming that the components of the NMF are the edges of the conical hull of all column vectors in the data matrix. Separable NMF was first shown to be tractable by [Arora, et al., 2012] [3]. [Donoho and Stodden, 2003] [7] also showed that separability implied that the factorization is unique, modulo scaling and permutation. The near separability assumption is satisfied in many natural applications. The separability condition can be interpreted as the pure-component assumption where it is assumed that there exists a column in the data matrix that comprises of exactly one component in the factorization for every component in W.

1.4 NMF applied to Music Decomposition

1.4.1 Existing Techniques

One of the most popular algorithms in use is a multiplicative variant of gradient descent first introduced by Lee and Seung [13] where W and H are alternatively optimized with the other variable kept constant. The algorithm converges to a local optimum but is fast and relatively easy to implement.

$$H_{kj} \leftarrow H_{kj}\frac{(W^TM)_{kj}}{(W^TWH)_{kj}}$$

$$W_{ik} \leftarrow W_{ik}\frac{(MH^T)_{ik}}{(WHH^T)_{ik}}$$

The advantage with multiplicative rules is that any musically motivated constraint can be hard-coded into the initialization matrix of W and H. NMF itself is underspecified and not unique, and the components identified in W, called the *Template Matrix* (H is called the *Activation Matrix*) may not correspond neatly to individual pitches/expected groups. Therefore, globally applicable information (e.g., equal temperament tuning to constrain W) as well as available information specific to the piece being analyzed (e.g., sheet music to constrain H) can be exploited [[18], [10], [1]] to extract a musically meaningful decomposition.

The downsides to this method are the fact that to extract meaningful components, one needs additional information to constrain the initializing matrices.

1.4.2 The Case for Separable NMF

By constraining the spectral signatures in the Template matrix to be signatures already found within the spectrogram, we can be sure that the signatures identified are legitimate and meaningful patterns that do occur in the piece of music. However, the flip side to this is the fact that the music may not satisfy the separability or the *pure-signal* assumption, i.e., all meaningful components may not occur at isolated points in time by themselves. However, for music with a lot of solo tracks that do satisfy this assumption to a greater degree, this method is promising. However, even for music that is very texturally rich at all points, the decomposition is sure to yield meaningful patterns, if not the notes by themselves.

Also, as a potential fix to the problem, appending the frequency signature samples of the instruments played solo to the spectrogram matrix (M) would decrease the residual by providing more potential vectors for the basis set while still keeping the final decomposition meaningful.

Chapter 2

Existing Algorithms for Separable NMF

Here we mention some algorithms for the near-separable NMF problem.

AKGM This was the first algorithm (by Arora, et al. 2012 [3]) to solve the separable NMF problem which proved that the separable NMF problem is tractable. It uses n feasibility LP formulations to obtain a provably good factorization for near-separable matrices under a bound on the $(\infty, 1)$ norm of the noise. However, since each each LP involves $O(n)$ variables, it cannot be scaled up to large datasets.

Hottopixx Another LP-based method with provable guarantees that involves $O(n^2)$ variables[Bittorf et al. 2012 [4]]. Hottopix was shown by [Chayan, et al. 2017 [6]] to perform poorly in noisy data due to its use of the $(\infty, 1)$ norm to define the simplex for the LP which is sensitive to outliers. Also, the value of ϵ (i.e., the upper bound for the $\| \cdot \|_{\infty,1}$ norm constraining the LP formulation) has to be known beforehand.

LP formulation LP ALCD An LP-based method similar to hottopixx was proposed by [Chayan, et al. 2017 [6]]. This algorithm overcomes the limitations of hottopix by moving the norm constraint into the objective function and using the $\| \cdot \|_{1,1}$ instead. The separability constraint, as in hottopixx, can be reformulated in terms of a *Factorization Localizing Matrix* $C_{(n \times n)}$ as:

$$
\begin{aligned}
M = WH &= M[I_r, 0]\Pi H \\
&= MC
\end{aligned}
$$

The Factorization Localizing Matrix is obtained by solving the following LP:

$$
\begin{aligned}
&\min_{\forall C \in R_+^{n \times n}} \|M - MC\| \\
&\text{subject to: } c_{ij} \leq a_i \quad \forall i, j
\end{aligned} \tag{2.1}
$$

$$\sum_{i=1}^{n} a_i \leq r$$

The columns selected by the algorithm, then, are those that correspond to the indices of the rows in C with the highest row-sum. This algorithm has been shown to be more robust to noise under certain conditions in [6], however solving the LP still takes a considerable amount of time on large datasets.

XRAY XRAY (Exterior Ray Projection Algorithm) [Kumar, et al., 2013 [12]] is a fast conical hull algorithm for separable NMF that finds the end-members that form the conical hull of the column vectors in the data matrix. It is quite fast and robust to noise. However, it can fail to identify the right components in the rare case that there are more than two columns that maximize the selection criterion. There are a number of variants with different but similar criteria to pick the next column to be included in the basis set. The remaining columns are then projected on to the conical hull of the basis columns to compute the residue ($M-$ projection of M on basis columns), which is then used to select the next column to be included, and so on.

Algorithm 1: XRAY [12]: Fast Conical Hull Algorithm

```
Input  : A near $r$-separable matrix $M = \tilde{M} + \eta$, where $\tilde{M} = WH$, $W = \tilde{M}_{:\kappa}$ for some
         $r$-subset $\kappa$, $W$ is an $\alpha$-simplicial, and $\|\eta\|_{\infty,1} \leq \epsilon$, and a factorization rank $r$
Output: Matrix $W$ and $H$ such that $M(:,\kappa) approx W$ for some index set $\kappa$ and $M \approx WH$
Initialize $R \leftarrow M, \kappa \leftarrow \{\}$.
while $|\kappa| < r$ do
  Detection Step: Find an extreme ray.
    $j^* = \arg\max_j \frac{R_{:i}^T M_{:j}}{p^T M_{:j}}$   for any $i : \|R_{:i}\|_2 > 0$
    where $p$ is a strictly positive vector (not collinear with $R_i$)
  Selection Step: Pick an exterior point using one of the following criteria:
      $rand$: any random $i : \|R_{:i}\|_2 > 0$.
      $max$: $i = \arg\max_k \|R_k\|_2$.
      $dist$: $i = \arg\max_k \|(R_k^T M)_+\|_2$.
  Greedy variant for choosing extreme ray: $j^* = \arg\max_j \frac{\|(R^T M_{:j})_+\|_2^2}{\|M_{:j}\|_2^2}$ .
  Update $\kappa \leftarrow \kappa \cup \{j^*\}$.
  Projection Step: Project onto the cone:-
      $H \leftarrow \arg\min_{B \geq 0} \|M - M_{:\kappa} B\|_2^2$.
  Update residuals: $R \leftarrow M - M_{:\kappa} H$.
end while
```

SNPA The Successive Nonnegative Projection Algorithm [Gillis, 2014] [8] is a modified version of the earlier Successive Projection Algorithm (SPA), first proposed by [Araùjo, 2001, [2]] which selects a column that maximizes a strongly convex function on the residue and then projects all remaining columns to the orthogonal complement of the selected vector. SNPA, instead, projects the remaining vectors to the convex hull of the columns and the origin to compute the residue. While SPA requires the columns of W to be of full rank, SNPA does not, and SNPA was also shown to be more robust to noise and applicable to a wider class of problems.

Algorithm 2: SNPA[Gillis, 2014] [8]: Successive Non-negative Projection Algorithm

```
Input  : A near $r$-separable matrix $M = \tilde{M} + \eta$, where $\tilde{M} = WH$, $W = \tilde{M}_{:\kappa}$ for some
         $r$-subset $\kappa$, and $\|\eta\|_{\infty,1} \leq \epsilon$, a factorization rank $r$, and a strongly convex
         function $f$ satisfying certain assumptions (in this study, the L2 norm)
Output: A set of indices $J$ such that $M_{(} : J) \approx W$ up to permutation
  Let $R = M$, $J = \{\}$, j=1.
while $R \neq 0$ and $j \leq r$ do
    $k^* = \arg\max_j f(R_{:j})$
    $J = J \cup \{k^*\}$.
    $R_{:i} = M_{:i} - M_{:J} H^*_{:i}$ for all $i$
        where $H^*_{:i} = \arg\min_{h \in \Delta} f(M_{:i} - M_{:J} h)$
    $j = j + 1$
end while
```

Chapter 3

Greedy Variants

We consider three variants of the greedy heuristic proposed in XRAY (however, we project the remaining columns to the convex hull of the basis columns and the origin in SNPA and not to the conical hull of the basis columns as in XRAY). The heuristic is based on the greedy strategy of trying to have the maximum possible sum of components of the residues along the new edge (basis column) to be chosen. This is an indicator of the magnitude of the drop in the objective (L2 norm of residue) to be minimized. By projecting every column of the residue matrix along every candidate basis column, we try to find the best candidate column that covers as much of the residue as possible. This greedy stepwise optimal strategy, however, is not guaranteed to output the right columns, even in the noiseless case. For example, if 4 points in 3 dimensions in the planar region $x + y + z = 0; x, y, z \geq 0$ are the vertices of an equilateral triangle and its center, then the matrix of column vectors that correspond to the position vectors of these points clearly supports a 3-separable factorization (with the 3 vertex points (column vectors) chosen as the basis columns), as the center can be expressed as a convex combination of the three vertices. However, the best 1-rank factorization (1-point approximation) would consist of just the point in the center of the triangle as part of the W matrix. The greedy heuristic would pick this point (column vector) in the first step and even if the next points it picks are on the vertices, since it does not drop a column (point) once picked, it returns an incorrect solution when the example clearly supports an exact 3-separable factorization.

Naturally, to attempt to try and remedy this shortcoming, we look at the possibility of dropping columns. Since the first column chosen tends to be near the *center* of the data, we could drop the chronologically first column and replace it with a new column picked using the same heuristic as before. In our preliminary analysis, the columns picked by this variant, which we call greedyRepSNPA (for replacement), were identical to the columns picked by SNPA in the noiseless case. However, even this proved to be futile.

In the example above, i.e., figure 3.2, the vertices of the pentagon in the planar region $x+y+z = 0; x, y, z \geq 0$ form the edges of the conical hull of the collection of points, however a sufficiently dense cluster of points near one of the vertices forces greedy to pick the first point within the cluster. The dense line (not as dense as the circular cluster) forces the next point to be picked as the endpoint of the line making the first *two* points picked by the greedy heuristic to be incorrect. Thus, we see that, even with replace, greedy cannot guarantee to obtain the right columns even in the noiseless case.

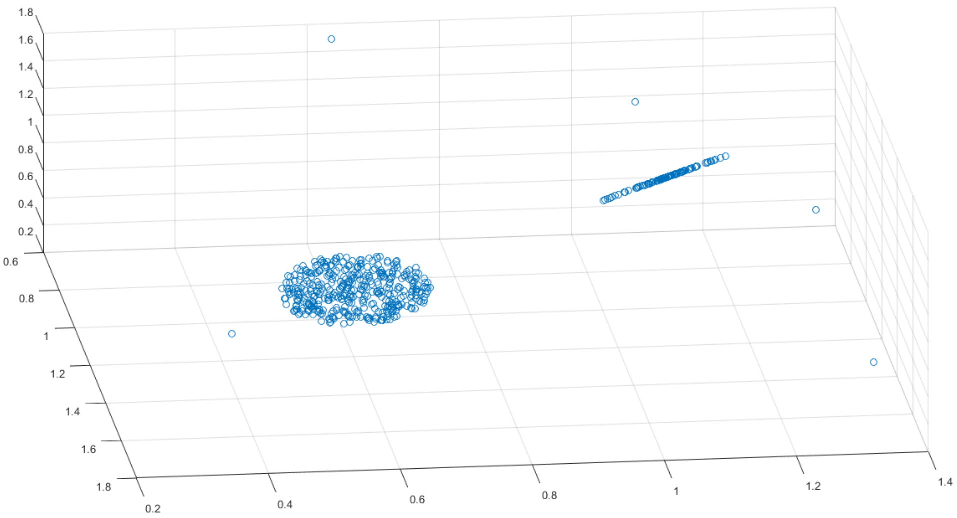

Figure 3.1: A noiseless example where greedyRepSNPA fails

What if we replaced the first two points instead of just the first? The counter example given above in Fig. 3.2 can be easily extended and modified to deal with any number of replacements, including an entire second pass, replacing every column in the basis. Though this algorithm (greedy2passSNPA) is not guaranteed to give the right columns, it is a good approximate solution most of the time.
When the data is sufficiently noisy, however, greedySNPA is comparable to the other algorithms even in column recovery and not just the residual.

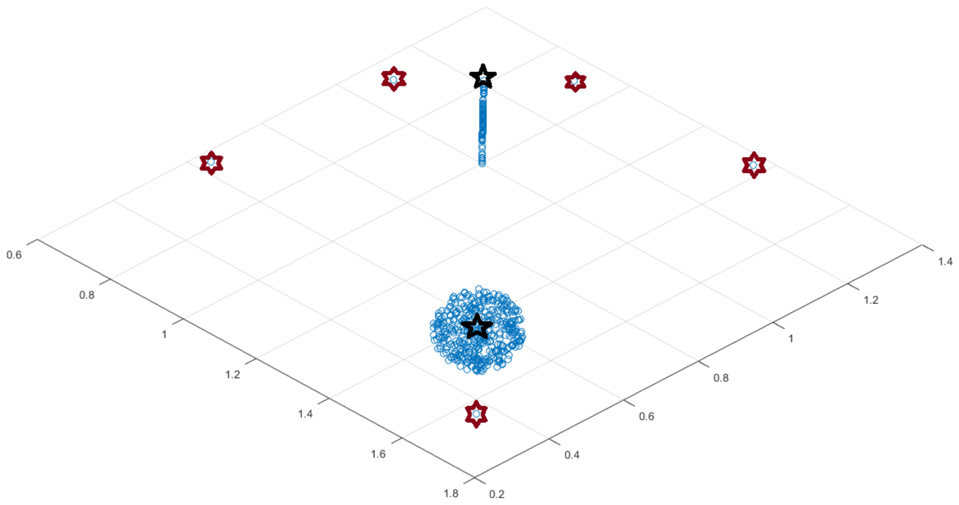

Figure 3.2: Rotated view of above image

3.1 Other Strategies

We know that the stepwise-optimal strategy using the L2 norm doesn't work without dropping columns, so what if we could instead start with the complete convex hull and drop columns greedily instead? Could we achieve a factorization whose distance with the optimal is no more than some constant (Here, distance is measured in terms of the $L_{2,2}$ matrix norm)? Although this method is guaranteed to converge in the noiseless case. The answer, unfortunately, turns out to be no.

In figure 3.3 we see that convex hull of all the points are the vertices of the trapezium in light green. Now, the best 3-point factorization would include the two vertices in the bottom and the midpoint of the upper edge of the trapezium (dark green) that form a triangle enclosing all the internal points in red. Clearly, this is not achievable by dropping points from the overall convex hull. Also, the density of the points (in effect, the number of points) clustered close to the vertices (in red) of the triangle can be increased arbitrarily to widen the gap between the output of the algorithm and the optimal factorization without any bounds that involve the $L_{2,2}$ norm. It may be possible to bound the error with an $L_{\infty,2}$ norm, however, which would be an interesting direction in the future.

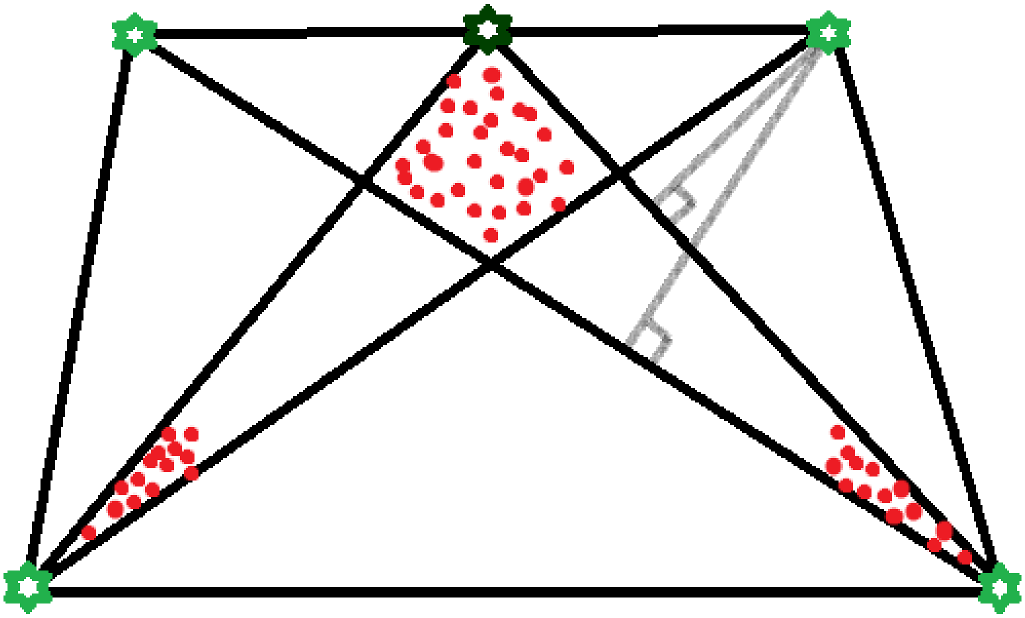

Figure 3.3: Counterexample for greedy dropping strategy

Chapter 4

Performance on Synthetic Data

We have compared the performance of six algorithms (LP ALCD, XRAY, SNPA, greedySNPA, greedyRepSNPA, and greedy2passSNPA) in this study and have compared the performance on different kinds of data:

4.1 Data Generation:

Generation of W: The columns of W are drawn from a uniform distribution $U(0,1)$. With just these columns, the data is well-conditioned. To generate data for the Ill-conditioned case, The SVD of the matrix obtained above is taken $W' = U\Sigma V^T$ and the diagonal matrix Σ is replaced with S where diagonal entries in S are of the form α^{i-1} for $i = 1, 2, ..., r$ such that $\alpha^{i-1} = 1000$. The condition number of this matrix is, thus, 1000. The negative entries in the new W are now made 0. The data is then L_1 (column) normalized.

Generation of H For the Dirichlet Distribution, a sample H' is drawn from a Dirichlet distribution where the parameters are chosen uniformly in $(0,1)$. The actual H is then computed as $H = [I_r I_r H']$ such that the basis columns are repeated once in the data matrix.

In Middle-Points generation $H = [I_r H']$ and each column in H' is just all zeros except two of the rows that have 0.5 in each entry. So, all the columns in the data matrix except the basis columns are midpoints between a pair of basis columns. Thus there are $\binom{r}{2}$ columns in H'.

Noise For the basis, sample, dimension, and sparsity variations as well as for the Dirichlet distributions, the noise added is δ times η $N(0,1)$ while for the Middle Points variation, the noise is $\eta_i = \delta(M_i - \bar{W})$.

4.2 Performance measures:

1. **Fraction of Columns extracted:-** Fraction of correctly extracted basis (columns of W).

$$\frac{\text{Number of correctly extracted columns of W}}{\text{Total number of columns of W (= r)}}$$

2. **Normalized Residual (L1):-**

$$\frac{\|M - M_{:J}H\|_{1,1}}{\|M\|_{1,1}} = \frac{\|M - WH\|_{1,1}}{\|M * scale\|_{1,1}}$$

3. **Normalized Residual (L2):-**

$$\frac{\|M * scale - M_{:J}H * scale\|_F}{\|M\|_F} = \frac{\|M * scale - WH * scale\|_F}{\|M * scale\|_F}$$

where $scale = diag(\frac{1}{sqrt(sumAlongColumns(M.^2)})$ so that $M * scale$ is l_2-normalized along the columns.

4. **Normalized Absolute Error:** $L_{1,1}$ norm of difference between original Factorization Localizing Matrix C and the one returned by the algorithm

$$\frac{\|C_{retrieved} - C_{original}\|_{1,1}}{\|C_{original}\|_{1,1}}$$

Performance measures for sparsity:

5. **Precision:** The precision measures the quality of the retrieved coefficients

$$\frac{|S_{retrieved} \cap S_{original}|}{|S_{retrieved}|}$$

6. **Recall:** The recall measures the quantity of retrieved coefficients

$$\frac{|S_{retrieved} \cap S_{original}|}{|S_{original}|}$$

where $S \in \{0,1\}^{n\times n}$ and $s_{ij} = 1, if c_{ij} \geq \alpha and 0 otherwise$.Here $|(\cdot)|$ is the number of zero-valued entries in $(\cdot)$ and intersection gives the number of common zeros between to sets.

4.3 Results

In the uniformly generated data sets for basis, dimension, and sample variations, the greedy2pass variant and the LP ALCD algorithms outperform every other algorithm in almost every parameter and are virtually indistinguishable from one another in terms of the performance metrics. In the Uniform Dirichlet, Uniform Middle-Points, Ill-Conditioned Dirichlet, and Ill-Conditioned Middle-Points datasets, however, greedy2pass SNPA slightly lags behind LP ALCD, XRAY, and SNPA.

Surprisingly, however, in both the Middle-Points datasets, as δ (noise) approaches 1, plain-vanilla greedySNPA surpasses every other algorithm. This effect is pronounced and markedly visible in the fraction of columns plots in figures of Ill-conditioned and Uniform Middle Point cases. This is probably because of the fact that even if many of the middle points are pushed away from the center of the conical hull, greedySNPA still tries to find the most *central* points in each iteration and is less affected by the skew.

Chapter 5

NMF applied to audio signals using STFT

Musical signals are dynamic, i.e., their statistics change over time. It would be rather meaningless to compute a single Fourier transform over an entire 10-minute song.

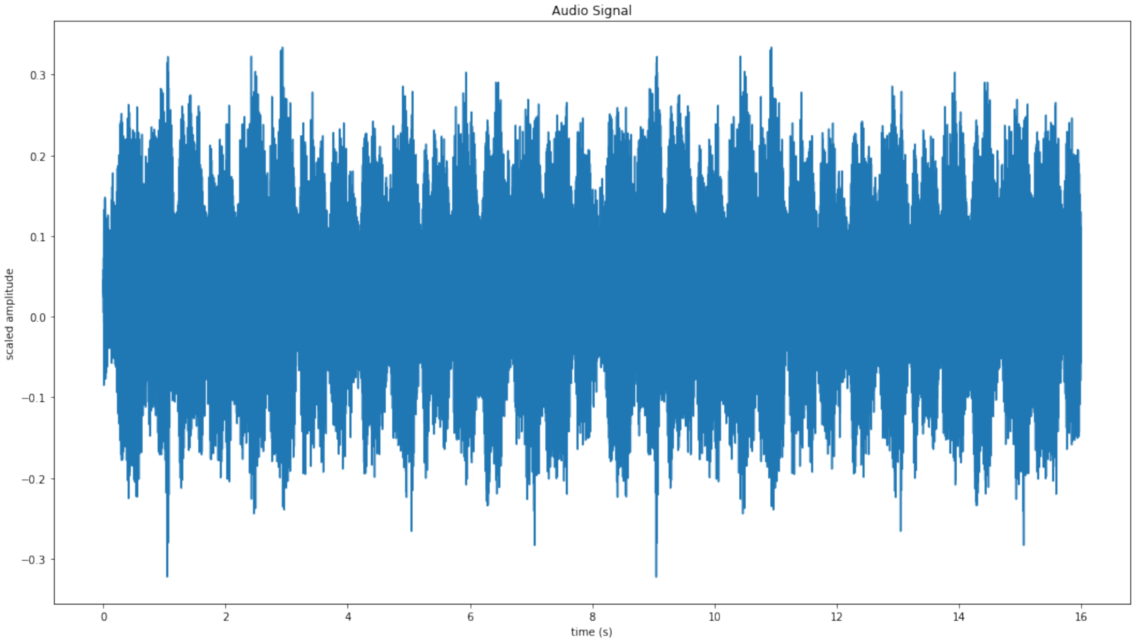

Figure 5.1: The audio signal x

To capture the variation of energy distribution across pitch with time, we compute the Fourier Transform of the original signal with a masking window function that is shifted across time. This results in a Matrix $X_{M+1 \times K+1}$ indexed by (m, k), the time and frequency indices respectively.

This is called the **short-time Fourier transform (STFT)** [14] and is defined as follows.

$$X(m,k) = \sum_{n=0}^{N-1} x(n+mH)w(n)e^{-2\pi jnk/N}$$

Here, x is the original signal, N is the length of the window (support of w), and H is the **hop-length**. Now, if the sampling frequency (or **sampling rate (sr)**) of the original signal is F_s (**sampling period** $= T_S = \frac{1}{F_S}$), then the physical frequency corresponding to coefficient k is:

$$F_{coeff}(k) = \frac{kF_s}{N}$$

The physical time corresponding to coefficient m is:

$$T_{coeff}(m) = \frac{mH}{F_s}$$

Now,

$$M = \lfloor \frac{T}{HT_s} \rfloor$$

where T is the duration of the signal.

And,

$$K = \lfloor \frac{N}{2} \rfloor$$

where N is the window length

So the frequencies represented in the spectrogram range in $[0, \frac{F_s}{2}]$

The matrix M used for NMF is obtained by taking the absolute value of the STFT on the audio signal and mapping the intensities of each bin in the resulting spectrogram into a component of a spectrum-vector. The spectrum vectors of each time interval are grouped together as column vectors to give M. W can be interpreted as the spectral signatures of a few fundamental notes/clusters of notes and H contains the temporal information for when each of these notes is “sounded”.

Now, each individual component can be converted back into the audio domain by computing an outer product of its corresponding column (frequency signature) in W and the corresponding row in H (time activations). The original phase is then recombined and an inverse STFT transform (ISTFT) is applied. Since it is not true, in general, that every spectrogram has a signal whose STFT is the spectrogram itself, the ISTFT is obtained by minimizing the norm of the error obtained in this process. For more details regarding the mechanism of this optimization, refer [9].

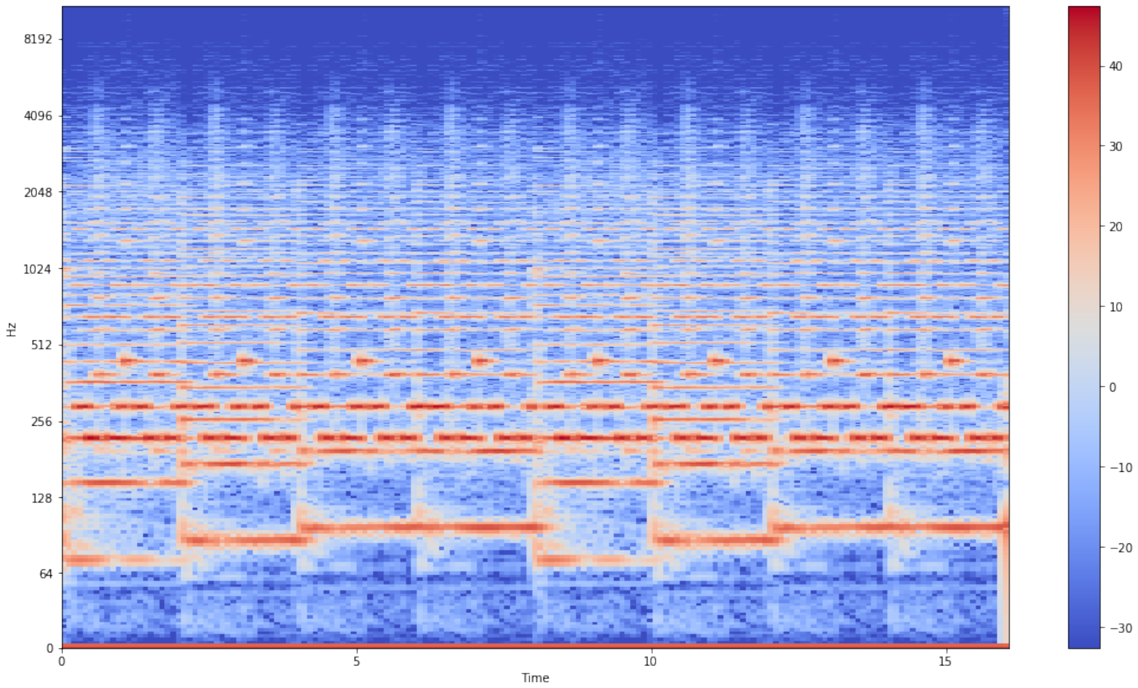

Figure 5.2: The spectrogram of the signal with intensities shown in dB

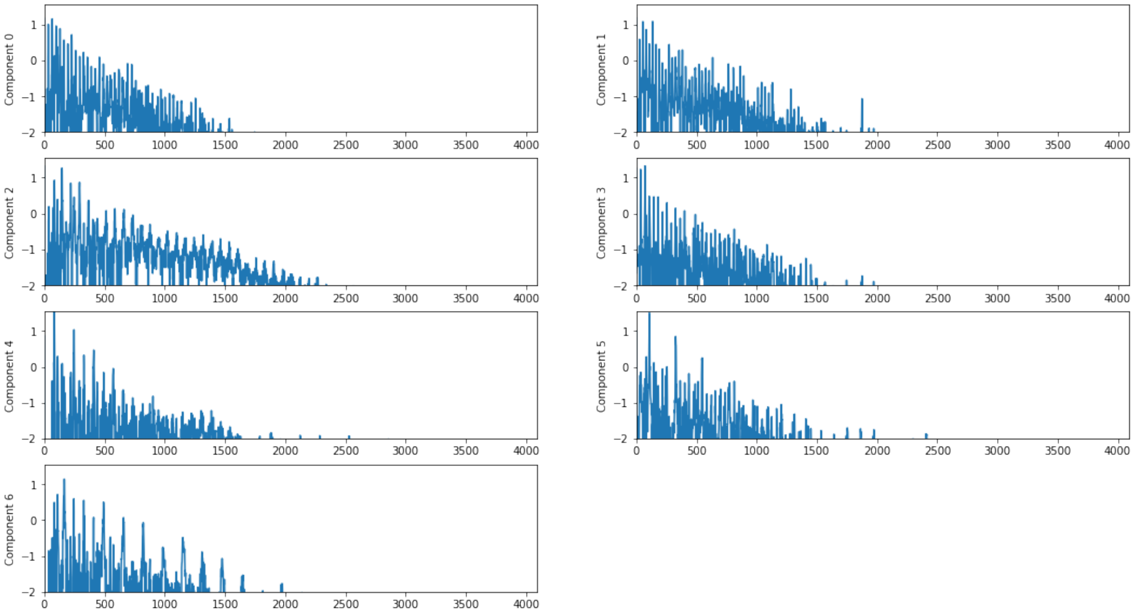

Figure 5.3: The spectral signatures of the components

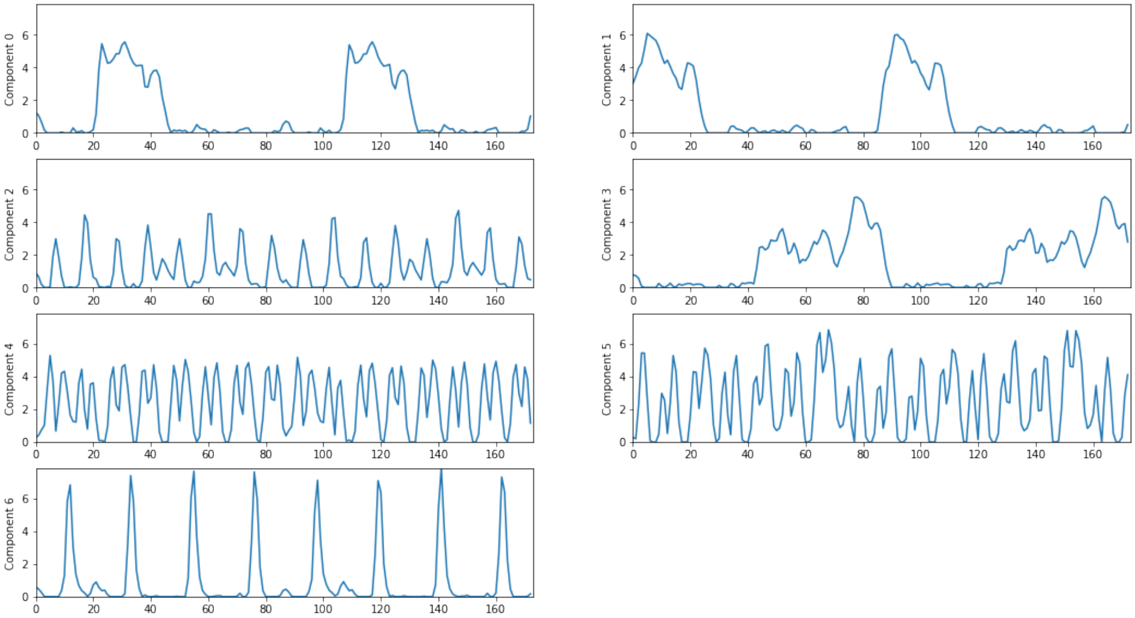

Figure 5.4: The time activations of the components

Chapter 6

Inferences

Three algorithms [NMF-Multiplicative Updates, sepNMF-SNPA, and sepNMF-Greedy (the 2-pass variant discussed above)] were tested with a mixed music sample containing the voices of a piano, a clarinet, and strings (7 different pitches sounded on 3 instruments) recorded digitally on GarageBand, a Digital Audio Workstation. The parameters for the STFT were as follows: `fft_length = length_of_window_function = 4096, hop_length = win_length/4 = 512, window_function used: Hann Filter`. The extracted components were then matched to the original sources by comparing their Euclidean distances, and the Signal to Noise Ratios (SNRs) were then computed with respect to the best match.

The perceptive quality of the separated component signals was good enough to be able to manually identify them as separate instrumental tracks and even identify pitches sounded, though the SNR scores are not very high (min SNRs were 1.3dB, -0.6dB, and 0.4dB, respectively for the three algorithms). The multiplicative updates rule recovered samples that best matched the original sources in this sample with SNR scores as high as 5.9dB. The reconstruction error of the three techniques compared are comparable, with the greedy algorithm doing better than SNPA, but not as well as multiplicative updates.

The sample was a combined track with few solo sections and was not approximated well with the pure signal model that sepNMF relies on. Different pitches are separated into different classes, despite being sounded by the same instrument. And while this has applications in note identification and automated music transcription, for the problem of source separation, where the objective is to separate the sounds of the different instruments, a model that learns to identify pitch-invariant patterns could be a better fit as discussed below.

Chapter 7

Future Work

Despite the overall signal reconstruction being of very good quality (with the potential of being used as a dimensionality reduction technique), the separation into sources is still poor, in reconstruction quality. The sources are separated by pitch-energy density signatures with a (nonnegative) linear mixing model. This does not take into account timbre, an essential aural characteristic that enables us to tell apart different instruments sounding the same pitch. Timbre is often thought to correspond to the energy distribution pattern across harmonics. While NMF uses this feature, it is not applied in a pitch-invariant fashion. The fact that the pattern is location-invariant in frequency (pitch invariant) naturally suggests that the pattern can be identified and learnt through the use of neural networks, particularly convolutional neural networks in a supervised fashion.

Supervised Source Separation There have been quite a few recent developments in guided and data-driven source separation that employ deep learning architectures. More recently, Chandna et. al [5] have proposed a low-latency convolutional neural network architecture to for this problem that outputs a time-frequency soft max mask for the STFT (similar to NMF) to reconstruct the sources given a mixed track.

Bibliography

[1] R. BLOUET J.L. DURRIEU A. OZEROV, C. FEVOTTE. Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, Czech Republic*, 2011.

[2] Saldanha B. Galvão R. Yoneyama T. Chame H. Visani V. Araùjo, U. The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. *Chemometrics and Intelligent Laboratory Systems*, 2001.

[3] Ge Rong Kannan Ravi Arora, Sanjeev and Ankur Moitra. Computing a nonnegative matrix factorization – provably. *STOC*, 2012.

[4] Recht Benjamin Re Christopher Bittorf, Victor and Joel A. Tropp. Factoring nonnegative matrices with linear programs. *NIPS*, 2012.

[5] Miron M. Janer J. Gomez E. Chandna, P. Monoaural audio source separation using deep convolutional neural networks. *13th International Conference on Latent Variable Analysis and Signal Separation*, 2017.

[6] Rahul Garg Chayan Sharma. 2016.

[7] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *NIPS*, 2003.

[8] Nicolas Gillis. Successive nonnegative projection algorithm for robust nonnegative blind source separation. 2014.

[9] D. W. Griffin and J. S. Lim. Signal estimation from modified short-time fourier transform. *IEEE Trans. ASSP*, 32:236–243, 4 1984.

[10] S. SAGAYAMA H. KAMEOKA, T. NISHIMOTO. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

[11] M. D. Plumbley J. Fritsch. Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), in , Vancouver, Canada*, 2013.

[12] Sindhwani V. Kambadur P. Kumar, A. Fast conical hull algorithms for near-separable nonnegativematrix factorization. *International Conference on Machine Learning*, 2013.

[13] Lee and Seung. Algorithms for non-negative matrix factorization. *NIPS*, 1999.

[14] Meinard Mueller. *Fundamentals of Music Processing.* Springer, 2015.

[15] Paatero and Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 1994.

[16] S. A. Raczynski; N. Ono; S. Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2007.

[17] Vavasis. On the complexity of non-negative matrix factorization. *SIAM Journal on Optimization*, 2009.

[18] N. Bertin; R. Badeau ; E. Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *Transactions on Audio, Speech, and Language Processing*, 2010.