NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES
INFORMATION AND COMMUNICATION TECHNOLOGIES

Master's Thesis

Singing Voice Separation from Monaural Recordings using Archetypal Analysis

Maria Eleni P. Sinni

ATHENS

OCTOBER 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

Διπλωματική Εργασία

Διαχωρισμός Τραγουδιστικής Φωνής από Μονοφωνικές Ηχογραφήσεις με την χρήση της Αρχετυπικής Ανάλυσης

Μαρία Ελένη Π. Σίννη

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2022

Masters Thesis

Singing Voice Separation from Monaural Recordings using Archetypal Analysis

Maria Eleni P. Sinni
A.M.: ic1180013

SUPERVISOR: Ioannis Panagakis, Associate Professor, NKUA

EXAMINATION COMMITTEE:

Vassilios Katsouros, Director ILSP, Athena Research Center

George Alexis Ioannakis, Assistant researcher, Athena Research Center

OCTOBER 2022

Διπλωματική Εργασία

Διαχωρισμός Τραγουδιστικής Φωνής από Μονοφωνικές Ηχογραφήσεις με την χρήση της Αρχετυπικής Ανάλυσης

Μαρία Ελένη Π. Σίννη
Α.Μ.: ic1180013

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Ιωάννης Παναγάκης, Αναπληρωτής Καθηγητής, ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Βασίλειος Κατσούρος, Διευθυντής ΙΕΛ, Ερευνητικό Κέντρο Αθηνά

Γεώργιος Αλέξης Ιωαννάκης, Ερευνητής Γ', Ερευνητικό Κέντρο Αθηνά

ΟΚΤΩΒΡΙΟΣ 2022

ABSTRACT

Singing voice separation aims at separating the singing voice signal from the background music signal from music recordings. This task is a cornerstone for numerous MIR (Music Information Retrieval) tasks including automatic lyric recognition, singer identification, melody extraction and audio remixing. In this thesis, we investigate Singing voice separation from monaural recordings by exploiting unsupervised machine learning methods. The motivation behind the employed methods is the fact that music accompaniment lies in a low rank subspace due to its repeating motive and singing voice has a sparse pattern within the song. To this end, we decompose audio spectrograms as a superposition of low-rank components and sparse ones, capturing the spectrograms of background music and singing voice respectively using the Robust Principal Component Analysis algorithm. Furthermore, by considering the non-negative nature of the magnitude of audio spectrograms, we develop a variant of Archetypal Analysis with sparsity constraints aiming to improve the separation. Both methods are evaluated on MIR-1K dataset, which is designed especially for singing voice separation. Experimental evaluation confirms that both methods perform singing voice separation successfully and achieve a value above 3.0dB in GNSDR metric.

SUBJECT AREA: Machine Learning and Signal Processing

KEYWORDS: Machine Learning, Signal Processing, Blind Source Separation, Singing Voice Separation, low-rank, sparseness

ΠΕΡΙΛΗΨΗ

Ο διαχωρισμός τραγουδιστικής φωνής στοχεύει στο να διαχωρίσει το σήμα της τραγουδιστικής φωνής από το σήμα της μουσικής υπόκρουσης έχοντας ως είσοδο μουσικές ηχογραφήσεις. Η εργασία αυτή είναι ένας ακρογωνιαίος λίθος για πλήθος εργασιών που ανήκουν στην κατηγορία ”ανάκτηση μουσικής πληροφορίας” όπως για παράδειγμα αυτόματη αναγνώριση στίχων, αναγνώριση τραγουδιστή, εξόρυξη μελωδίας και ρεμίξ ήχου. Στη παρούσα διατριβή, διερευνούμε τον Διαχωρισμό τραγουδιστικής φωνής από μονοφωνικές ηχογραφήσεις εκμεταλλευόμενοι μεθόδους μη επιτηρούμενης μηχανικής μάθησης. Το κίνητρο πίσω από τις μεθόδους που χρησιμοποιήθηκαν είναι το γεγονός ότι η μουσική υπόκρουση τοποθετείται σε έναν χαμηλής-τάξης υπόχωρο λόγω του επαναλαμβανόμενου μοτίβου της, ενώ το πρότυπο της φωνής παρατηρείται ως αραιό μέσα σε ένα μουσικό κομμάτι. Συνεπώς, ανασυνθέτουμε ηχητικά φασματογραφήματα ως υπέρθεση χαμηλής-τάξης και αραιών συνιστωσών, αποτυπώνοντας τα φασματογραφήματα της μουσικής υπόκρουσης και τραγουδιστικής φωνής αντίστοιχα χρησιμοποιώντας τον αλγόριθμο Robust Principal Component Analysis. Επιπλέον, λαμβάνοντας υπόψη τη μη αρνητική φύση του μέτρου του ηχητικού φασματογραφήματος, αναπτύξαμε μία παραλλαγή της Αρχετυπικής Ανάλυσης με περιορισμούς αραιότητας στοχεύοντας να βελτιώσουμε τον διαχωρισμό. Αμφότερες οι μέθοδοι αξιολογήθηκαν στο σύνολο δεδομένων MIR-1K, το οποίο είναι κατασκευασμένο ειδικά για τον διαχωρισμό τραγουδιστικής φωνής. Τα πειραματικά αποτελέσματα δείχνουν πως και οι δύο μέθοδοι εκτελούν τον διαχωρισμό τραγουδιστικής φωνής επιτυχημένα και πετυχαίνουν στην μετρική GNSDR τιμή μεγαλύτερη των 3.0dB.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Μάθηση και Επεξεργασία Σήματος

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Επεξεργασία Σήματος, Τυφλός διαχωρισμός πηγών, Διαχωρισμός Τραγουδιστικής Φωνής, χαμηλή-τάξη, αραιότητα

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

1. INTRODUCTION

1.1 Overview

Blind source separation (BSS) refers to the separation of a set of source signals from their mixed signal. Neither the initial sources nor the mixing procedure is a-priori known. Singing Voice Separation (SVS) is a subcategory of the broad BSS category that is aiming to separate the singing voice from the music accompaniment. Singing voice carries important information about the singer, the lyrics, the language, the genre and other characteristics of a song. So there is a need to remove or attenuate the background music because it acts like noise and that is confusing the information retrieval procedure. This challenging task is even more complex when the available sources are recorded by one microphone, i.e. monaural. The human auditory system has the ability to isolate sources and separate voice from music accompaniment even if the mix is monaural. Machines need to extract some features from a trained dataset or make some assumptions about the form of the song to perform separation. Existing methods that perform SVS are classified as supervised and unsupervised.

Supervised methods have a model for each source or for one of the sources (voice,music) and follow a method that classifies signals onto a feature space where the separation is performed e.g. pitch-based interference [4], [5], adaptive Bayesian modeling [6]. There are also deep learning methods that perform separation with very good results, as Huang et al. in [7], but they surpass our theoretical framework and we will not analyse them in this thesis.

On the other hand, unsupervised methods make some fundamental assumptions about song's nature that do not require prior labeled datasets and extracted features. The most popular methods used today for SVS are Independent Component Analysis (ICA) [8], [1], Non-negative Matrix Factorization [9], and Robust Principal Component Analysis (RPCA) [10]. Each one of them decomposes the signal using a different logic, for example ICA considers the input mixed signal matrix as a linear combination of the true sources and performs an algorithm that finds statistically independent components that each one corresponds to a true source. Non-negative matrix factorization (NMF) algorithms also perform audio source separation [11], [12] using a different methodology. NMF is a matrix decomposition method that performs separation in a two dimensional matrix X, resulting to a product of two non-negative matrices W, H such that their product is almost equal to the initial matrix. W consists of basis vectors and H their corresponding weights. Each column of the basis vector matrix represents a note event or a dictionary and each row of the weight matrix is a temporal envelope of the same note event or an activation coefficient. The goal of this method is to decompose the initial matrix into components that after multiplying them with the phase of the initial matrix, can reproduce the true sources.

Another approach is called Robust Principal Component Analysis (RPCA) and is based on the assumption that popular songs have a background that is mainly repeating itself so it can be expressed in a low-rank subspace, while the singing voice is considered sparse within a song and non-repetitive [2]. This is the baseline method for our experiment, which we reproduced and compared with our proposed method. In chapter 2.2.3 we will present the RPCA method and analyse its results.

In this thesis, we propose a model based on proven Robust PCA's assumptions, that is the separation of a song into low-rank and sparse representations. Our hypothesis is that besides its repeating nature, the background music is also non-negative, because the magnitude of a spectrogram is always non-negative, and also has other inherent characteristics (like melodic, rhythmic, morphological) that we want to capture. These characteristics are called archetypes and we believe that they will encapsulate the background with higher precision, in order that the rest of the song (i.e voice) will be more clear and as a result we will have better separation quality. The contribution of this thesis is a novel algorithm about singing voice separation.

1.2 Thesis structure

The present thesis is organized as follows. In Section 2, we introduce existing methods that perform singing voice separation and focus on Robust PCA method. In Section 3 we analyse our Proposed Method in detail. In Section 4, we present the results of our experiments using MIR-1K dataset. Finally, a concluding statement summing up the findings of this research and various ideas for system improvement are included in Section 5.

2. RELATED WORK

In this chapter, we present some existing algorithms for solving singing voice separation problem that will help us look into the problem and make comparisons. Early methods for SVS, as mentioned earlier, are separated in two main categories based on the existence of prior knowledge, namely supervised and unsupervised methods.

2.1 Supervised methods

Supervised methods perform pretraining techniques to the model before performing the separation. Firstly, they map signals into a feature space and then they identify the singing voice segments. After singing voice detection a source separation technique is applied. For example, some methods that are presented below are based on pitch detection or on algebraic properties.

2.1.1 Pitch-based interference methods

One way to explore and extract information from a song is to take advantage of the fact that vocal signals and some musical instruments are approximately harmonic. This means that they are composed of harmonic partials, i.e. positive integer multiples of the fundamental frequency of the sound. Pitch-based interference methods use the vocal pitch envelope as a clue to separate vocal harmonics from the musical background.

Li and Wang [4] proposed a computational auditory scene analysis (CASA) [13] system which uses a binary masking technique to separate the singing voice from the accompaniments. This method decomposes the mixed signals into sensory elements called time–frequency (T-F) units using an auditory filterbank. Afterwords, T-F units are labeled based on which label (voice,music) is dominant according to detected pitch contours. These detected pitch contours are then used for singing voice separation by grouping into T-F units by their harmonicity.

Hsu et al. [5] proposed a pitch-based inference system that detects unvoiced parts of the input signal, then separates them from the musical background and by combining this method with Li et al.'s method [4] managed to separate both voiced and unvoiced singing voice from the music accompaniment. First, they take a mixed input and detect Accompaniment/Unvoiced singing voice/Voiced singing voice (A/U/V) and make a time-frequency (T-F) decomposition. Next, the system identifies voiced-dominant T-F units within each voiced frame and unvoiced-dominant T-F units within each unvoiced frame and finally these unit are re-synthesised and construct the separated singing voice.

2.1.2 Adaptive Bayesian modeling

Another approach is exploiting algebraic properties, that is taking advantage of a training database by learning some parts of the model, in order to guide the estimation process into improved solutions. Ozerov et al. [6] proposed an adaptive Baysian model for single-channel separation. This model is based on the observation that the models of the source signals match precisely the statistical properties of the mixed signal. They developed this idea by adding the adaptation technique to the source signals with respect to the mixed

signals so that they train the model. They represent each source (vocals, accompaniment) with a Gaussian mixture model (GMM) and train the model to make source estimation. Afterwords, they use the Maximum a posteriori (MAP) adaption criterion [14] to adapt the general music model on the non-vocal parts of a particular song.

2.2 Unsupervised methods

The next source separation category does not use any a priori knowledge or particular features in order to train the model. Unsupervised methods make some fundamental assumptions about the song's structure and then choose the proper separation technique to classify the audio into individual components. In the case of music signals, each component usually represents a musically meaningful entity or parts of it, so that different entities are represented with different components. The entities can be for example the sounds produced by a percussive instrument or in our case, the music accompaniment and the singing voice.

Mathematically, assuming that a mixture signal $y(t)$ is composed of N sources, $x_n(t)$, for $n = 1...N$, such that

$$y(t) = \sum_{i=1}^{N} x_i(t),$$

the goal of an unsupervised source separation system is to recover one or more $x(t)$'s, given only $y(t)$. Now we will present some examples of unsupervised methods for SVS:

2.2.1 Independent Component Analysis (ICA)

Hyvärinen and Oja [8] developed Independent Component Analysis algorithm which is targeting on finding a linear combination of non-Gaussian data in order that the individual components are statistically independent. In BSS, in order to separate N signal sources, we must have at least N microphones. In the simple scenario (motivated from the Cocktail-Party Problem [15]), two sounds are generated by music and a voice and recorded simultaneously in two microphones.

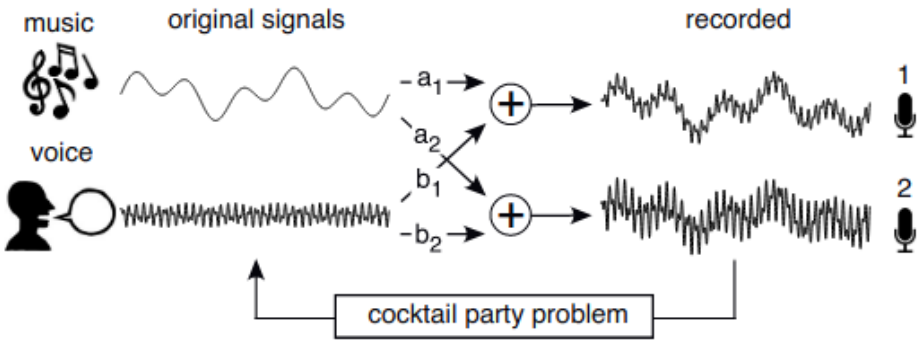


Figure 1: Example of cocktail party problem [1]

Let $s_1(t)$,$s_2(t)$ be the signals emitted by the two sources and the recorded signals are denoted by $x_1(t)$,$x_2(t)$. These recorded signals can be expressed as a linear combination of source signals:

$$x_1(t) = a_{11}(t)s_1(t) + a_{12}(t)s_2(t)$$
$$x_2(t) = a_{21}(t)s_1(t) + a_{22}(t)s_2(t)$$

where parameters in matrix A= $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ depend on distances of the microphones to the speaker, along with other microphone properties. With mixed sources $x_1(t)$,$x_2(t)$ given, the goal is to find an estimate about the matrix A and finally the true sources $s_1(t)$,$s_2(t)$. This can happen by finding the reverse A matrix and multiply it with the mixed sources.

ICA requires that the number of observed mixtures must be larger or equal to the number of sources. This requirement makes this algorithm be very specific in terms of audio source separation because most songs are usually mixed in one or two channels (mono or stereo), always less than the source instruments.

There are several ways of implementing ICA based on the contrast function that measures independence. For Blind Source Separation (BSS) tasks most researchers use an approximation of negentropy in an ICA version called FastICA [8].

2.2.2 Non-negative Matrix Factorization (NMF)

NMF is an unsupervised iterative algorithm that factorizes a nonnegative m by n matrix $X \in \mathbb{R}^{\geq 0, m \times n}$, into a product of two nonnegative matrices, $W \in \mathbb{R}^{\geq 0, m \times r}$ and $H \in \mathbb{R}^{\geq 0, r \times n}$ where $r \leq m$ is matrices' rank[1]. The goal is to minimize the distance (error of reconstruction) between X and the product WH:

$$min_{W,H} d(X, WH)$$

We could alternatively examine this factorization as reduced-rank basis decomposition such that:

$$X \approx WH$$

In the source separation task let input matrix X be a magnitude spectrogram and its rank r be the number of components of the decomposition. We target on finding matrices W and H such that the columns of W are the basis vectors (or features in the frequency domain) and each column of H represents the corresponding weights that vary over time.

For example, let's assume that we have an input mixed signal that consists of two source signals: one piano and one singer. We want to decompose the mixed signal X in a product of W and H. The result we get after the decomposition is one linear combination that corresponds to the piano component and one to the singer component. Each component consists of a subset of some columns of W multiplied by the corresponding subsets of H. Now if $S \subset \{1, \ldots, r\}$ then the two parts are:

[1]The quality of the separation procedure depends directly by the selection of rank parameter r.That means that if we choose an appropriate rank, we will probably have a very good estimate of the target sources.

$$V_{piano} = \sum_{i \in S} \mathbf{w}_i \mathbf{h}_i^T$$

$$V_{singer} = \sum_{i \in \{1,\ldots,r\} \setminus S} \mathbf{w}_i \mathbf{h}_i^T$$

In figure 2 we can observe the spectrogram/frequency representation of a simple piano compose where V is the magnitude spectrogram of the input signal, W_S is the subset of basis vectors and H_S is the activations properly chosen to reconstruct source S.

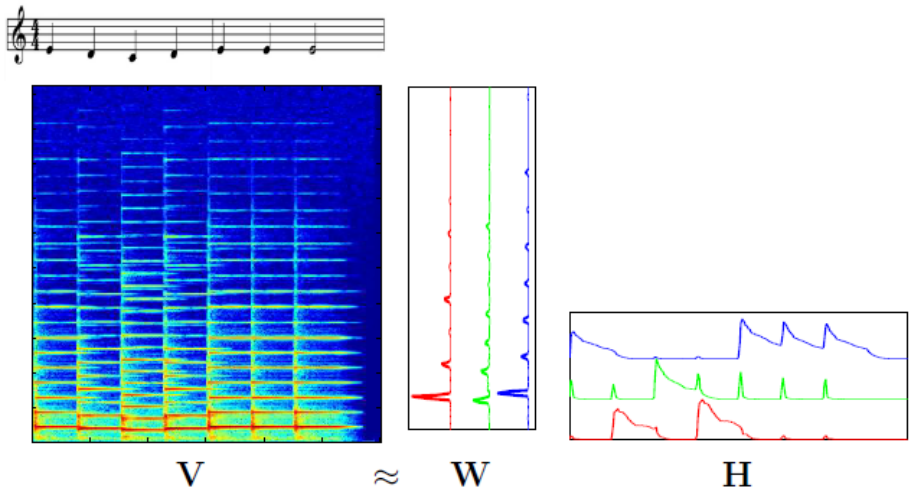


Figure 2: NMF of Mary Had a Little Lamb with piano for r=3

For example, Virtanen [11] presented an NMF algorithm for monaural source separation which combines temporal continuity with sparseness objectives. The algorithm is performed in pitched musical instrument mixed samples and percussion instruments. Firstly, the magnitude spectrogram of the input signal is factorized into a linear sum of components which are a basis function and its time-varying gain. Each components sum corresponds to a source and is non-negative. Then, the algorithm groups these components such that they form an individual source. (A source can be percussive instrument or a certain group of all pitch-notes of all instruments.) The components' gains are restricted to be slow-time varying sparse. Temporal continuity is measured by assigning a cost to large changes between gains and neighbour frames. This restriction is proved that improves the detection of pitched musical instruments, therefore the separation.

Smaragdis and Brown [12] presented a methodology about non-negative matrix factorization for polyphonic music transcriptions. They model the song content of the music transcription by a linear basis transform and perform NMF on the input mix matrix. By performing NMF, they decompose the input matrix in two matrices (W,H), one for spectral note events (W) and one for the envelopes (temporal activity) of these note events (H). The pairing of a group of W's columns with the corresponding rows of H describes a note event. Here, the choice of the matrices' rank plays a very important role in the quality of the results. If the rank is smaller, the analysis is incomplete. So a safer choice would be to choose a bigger rank. One of the shortcomings of this approach is that it requires music passages from instruments with notes that exhibit a static harmonic profile. They tend to address this issue in future with alternative decomposition methods that have more expressive power than linear transforms.

2.2.3 Robust Principal Component Analysis (RPCA)

In this chapter we will analyse Robust Principal Component Analysis algorithm, which is the baseline method and our inspiration for this thesis. Candes et al. [10] proved that a large data matrix M can be decomposed as:

$$M = L_0 + S_0 \tag{2.1}$$

where L_0 is low-rank and S_0 is sparse. The problem is formed using the $Principal\,Component\,Pursuit(PCP)$ approach:

Let $M \in \mathbb{R}^{n_1 \times n_2}$,$||M||_* = \sum_i |\sigma_i(M)|$ be the nuclear norm (sum of singular values; a convex relaxation of the rank of the matrix) of the matrix M and $||M||_1 = \sum_{i\,j} |M_{ij}|$ the l_1-norm (sum of absolute values; a convex relaxation of l_0-norm) of M. The formed problem is:

$$\begin{aligned} &\text{minimize} \quad ||L||_* + \lambda||S||_1 \\ &\text{subject to} \quad L + S = M \end{aligned} \tag{2.2}$$

where the parameter λ has the value $1/\sqrt{\max(n_1, n_2)}$, as suggested in [10].

By solving (2.2), a low-rank matrix and a sparse matrix are exactly recovered when the next two assumptions are satisfied:

1. low rank component is not sparse
2. sparse matrix is not low rank

Neither l_1-norm nor nuclear norm are separable functions. Problems that consist of non separable functions are solved using proximal operators [16].

To solve the convex PCP problem (2.2) writers transformed it using Augmented Lagrangian Multiplier (ALM) method [17]:

$$minimize \quad ||L||_* + \lambda||S||_1 + \langle Y\ ,\ M - L - S\rangle + \frac{\mu}{2}||M - L - S||_F^2 \tag{2.3}$$

If we apply ALM directly to the problem, the minimization will occur at both L and S simultaneously. This is a tough problem and does not take into account that the objective function is separable.

An optimal solution for this problem is Alternating Directions Method [18] which divides the problem in two sub-problems that minimize over L and S respectively. To achieve this the problem is transferred in an equivalent form:

$$\mathcal{L} = ||L||_* + \lambda||S||_1 + \frac{\mu}{2}||M - L - S + \frac{Y}{\mu}||_F^2 \tag{2.4}$$

so that we solve the individual:

$$\begin{cases} L_{k+1} = \arg\min\limits_{S}\ \mathcal{L}(L, S_k, Y_k), & \text{(2.5a)} \\ S_{k+1} = \arg\min\limits_{L}\ \mathcal{L}(L_{k+1}, S, Y_k), & \text{(2.5b)} \\ Y_{k+1} = Y_k + \mu(M - L_{k+1} - S_{k+1}). & \text{(2.5c)} \end{cases}$$

Let $\mathcal{S}_\tau : \mathbb{R} \to \mathbb{R}$ be the soft-thresholding (shrinkage) operator: $\mathcal{S}_\tau[x] = sgn(x)max(|x|-\tau, 0)$ where $x \in \mathbb{R}$ and $\tau > 0$. This operator can be extended to vectors and matrices by applying it element-wise. It is easy to show that:

$$\arg\min_S \mathcal{L}(L, S, Y) = \mathcal{S}_{\lambda/\mu}(M - L + \mu^{-1}Y) \tag{2.6}$$

Likewise let, $\mathcal{D}_\tau(X)$ be the singular value thresholding operator, where $\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^*$, where $U\mathcal{S}_\tau V^*$ denotes the singular value decomposition. We can easily show that:

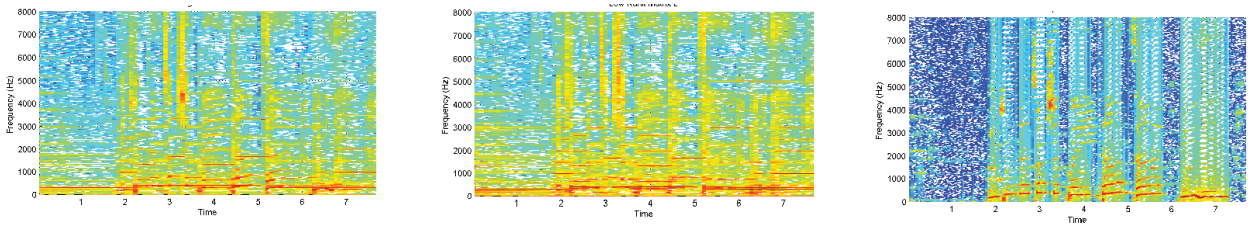
$$\arg\min_L \mathcal{L}(L, S, Y) = \mathcal{D}_{1/\mu}(M - S + \mu^{-1}Y) \tag{2.7}$$

Finally, the parameter Y is updated with the rule: $Y_{k+1} = Y_k + \mu(M - L_{k+1} - S_{k+1})$. To reduce the number of iterations the algorithm selected to solve the RPCA problem is called *"Inexact ALM"* [17]. This algorithm updates each unknown parameter once, and repeats until it converges.

Based on the assumption that the background music follows a repeating pattern, so it is considered low rank, and singing voice is sparse within the song, Huang et al. [2] proposed a method that uses RPCA algorithm to perform singing voice separation in monaural recordings. The music accompaniment is considered to be a low-rank matrix L and the singing voice a sparse matrix S such that $M = L + S$.

Firstly, they perform Short-Time Fourier Transform (STFT)(check Appendix A) to the input signal and take its magnitude to obtain the spectrogram denoted as matrix M. The spectrogram of each mixture is computed using a window size of 1024 and a hop size of 256.

Next, they apply the "Inexact ALM" algorithm with input the magnitude matrix $|M|$ in order to perform separation. This procedure resulted in two matrices, one for the background music part (L) and one for the vocals part (S). From the spectrograms in figure 3 we can observe the differences between the sparse part and the low-rank part:



(a) Original Matrix M **(b) Low Rank Matrix L** **(c) Sparse Matrix S**

Figure 3: Example RPCA results for yifen_2_01 at SNR=5 for (a)the original matrix, (b)the low-rank matrix, and (c)the sparse matrix. The figure is borrowed from [2]

After that, they restored the phase component to the magnitude spectrogram in order to get back the complex matrix form and performed Inverse Short Time Fourier Transform (ISTFT) (for more details check chapter 4.3).

For better separation results researchers experimented by applying a binary time-frequency mask at the mix. A mask is a matrix that is the same size as a spectrogram and contains values in the inclusive interval [0.0,1.0] and is defined as follows:

$$M_b(m,n) = \begin{cases} 1 & if\ |S(m,n)| > gain * |L(m,n)| \\ 0 & otherwise \end{cases} \tag{2.8}$$

for all $m = 1...n_1$ and $n = 1...n_2$

The mask is applied to the original STFT matrix M to obtain the two components, one for the singing voice and one for the accompaniment:

$$\begin{cases} X_{singing}(m,n) = M_b(m,n)M(m,n) \\ X_{music}(m,n) = (1 - M_b(m,n))M(m,n) \end{cases} \tag{2.9}$$

for all $m = 1...n_1$ and $n = 1...n_2$

The figure below describes the whole procedure from input mixed signal until the evaluation of RPCA model:

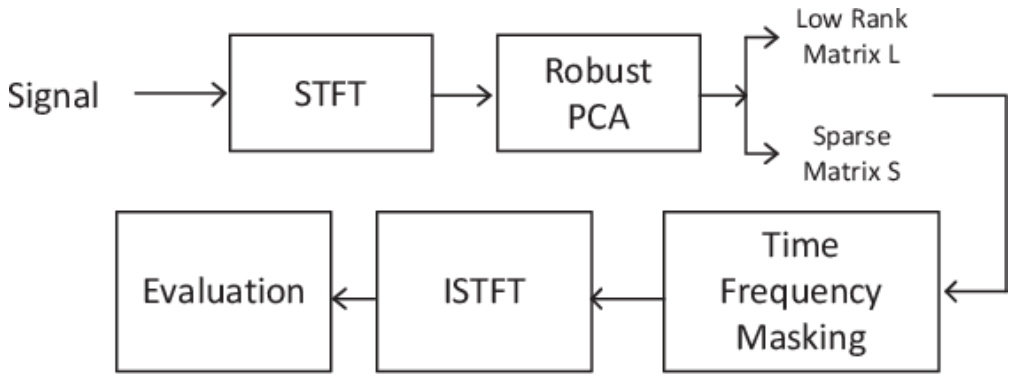


Figure 4: Robust PCA framework [2]

For the evaluation procedure writers used MIR-1K dataset (chapter 4.1) and mixed the clips for -5, 0 and 5 SNRs. The performance of the model was measured using the standard Blind Audio Source Separation (BASS) metrics: Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR) and Source to Artifacts Ratio (SAR) (chapter 4.4.1, [19]) and also Global Normalized SDR (GNSDR) metric [5], [20].

They experimented examining different gain values {0.1, 0.5, 1, 1.5, 2} (energy between the sparse matrix and the low-rank matrix), different $\lambda_k = k/\sqrt{\max(n_1, n_2)}$ values for $k = \{0.1, 0.5, 1, 1.5, 2, 2.5\}$ (the trade-off parameter between sparsity of S and the rank of L) and performed separation with and without mask.

They empirically chose λ_1 and gain = 1 to compare RPCA with other models. The results showed that singing voice separation using RPCA with and without mask achieved a very good performance and the comparison with previous systems confirmed this assertion. For SNR=0 they achieved a GNSDR value about 2.37dB without mask and 2.57dB with the use of ideal binary mask.

3. ARCHETYPAL ANALYSIS FOR SINGING VOICE SEPARATION

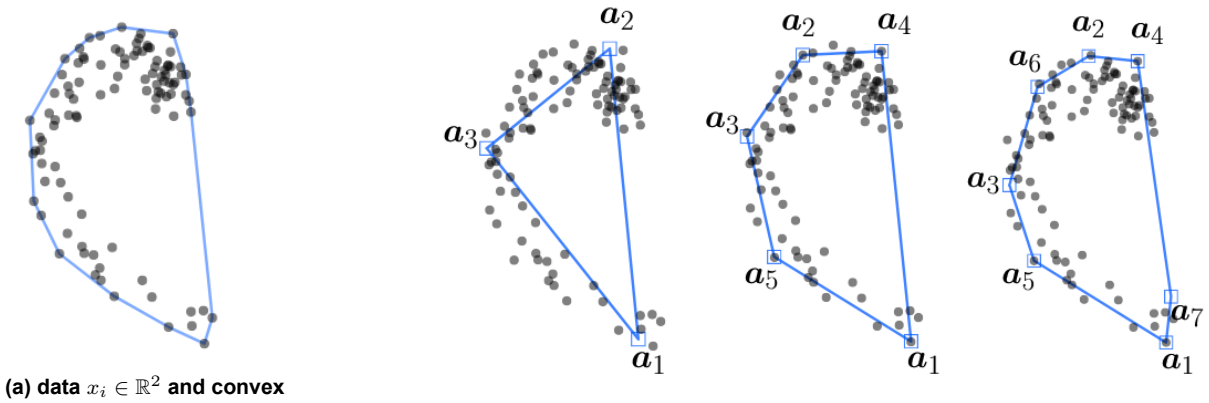
In this chapter we present our proposed methodology for singing voice separation from monaural recordings using Archetypal Analysis in detail. We analyse the theoretical framework of Archetypal Analysis method and how it can be implemented for blind audio source separation purposes and specifically for singing voice separation which is our objective in this present work. Based on Huang et al.'s work [2] in which SVS problem is solved using Robust Principal Component Analysis (RPCA), our innovation is the exploitation of signal's structure (format) in order to improve the separation of music accompaniment. This can happen if we introduce archetypes in the separation procedure which will produce better low rank matrix and consequently a better sparse matrix.

Singing voice separation problem has multiple solutions. One of the most popular is in the spectrogram factorization field and is about separating a matrix X into a low-rank matrix L, which represents the music accompaniment, and a sparse matrix S, which represents the singing voice. In previous works [2], low rank matrix was found using the nuclear norm. Alternatively, we can represent L as a product of two low-rank matrices if we know their rank.

By taking advantage of the fact that the magnitude of the spectrogram is non-negative by definition, we separate the low-rank matrix into a product of two low-rank and non-negative matrices C and S. We develop our idea by deploying more inherent characteristics of the background music, like repetition, melodic, rhythmic and morphological characteristics. These characteristics are called archetypes and in order to capture them we use Archetypal Analysis algorithm.

Archetypal analysis proposed by Cutler and Breiman (1994) [21] estimates the principal convex hull (PCH)[1] of a dataset. The convex hull of a set of points V is a polytope whose 'corners' contain all representative points of V (Figure 5). These points are identical to the data. By finding the convex hull we find the most informative samples of a dataset that describe all data as a linear combination of the informative ones. Convex hull does not find an arbitrary non-negative low-rank decomposition, but a decomposition that is adapted on the musical background. We expect to find these features that describe the low-rank part, so the rest of them will be adopted by the sparse part.

[1] A set of points in a Euclidean space is defined to be convex if it contains the line segments connecting each pair of its points.



(a) data $x_i \in \mathbb{R}^2$ and convex hull

(b) convex hulls for k = 3,4,7 archetypes

Figure 5: Archetypal Analysis computes the convex hull of the data. In this figure we can observe the significance of the number of k archetypes selection. More archetypes make the convex hull more informative [3]

3.1 Background

Let us consider a sample of n data points x_i in $\mathbb{R}^m$. We gather them in a data matrix

$$X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{m \times n},$$

and determine the number of archetypes $1 < k << min\{n, m\}$ and two column stochastic matrices

$$C = [c_1, c_2, ..., c_k] \in \mathbb{R}^{m \times k}$$
$$S = [s_1, s_2, ..., s_m] \in \mathbb{R}^{k \times m}$$

These two matrices form the matrix factorization problem:

$$\text{min} \quad ||X - XCS||_F^2 \tag{3.1}$$

Now we introduce matrix A such that $A = XC$ and rewrite X as:

$$X \approx XCS = AS$$

where the k column vectors a_j of the matrix $A \in \mathbb{R}^{n \times k}$ are the archetypes of data. Archetypal analysis has two properties:

1. Since $A = XC$ and C is a column stochastic matrix, **each archetype $\boldsymbol{a_i}$ is a convex combination of data points: $\boldsymbol{a_j \approx Xc_j}$,**

   where vector c_j corresponds to the j-th column of C

2. Since $X \approx AS$ and S is a column stochastic matrix, **each data vector $\boldsymbol{x_i}$ is a convex combination of archetypes: $\boldsymbol{x_i \approx As_i}$,**

   where vector s_i corresponds to the i-th column of S.

3.2 Proposed method: Archetypal analysis with sparsity constraints

Let X be the input spectrogram that represents the mix. We want to formulate a learning framework that is derived by the following optimization problem:

$$\begin{aligned} \min_{C,S} \quad & ||X - XCS||_F^2 \\ \text{s.t.} \quad & C \geq 0,\ S \geq 0,\ 1^T C = 1^T,\ 1^T S = 1^T, \end{aligned} \tag{3.2}$$

where the constraints ensure that the archetypes are convex combinations of the data points and that data points are well approximated by convex combinations of archetypes.

The matrix XCS represents the low-rank term that stands for the music accompaniment part we tend to improve. Problem (3.2) has to be reformed in order to contain the singing voice part, that is represented by the sparse matrix E.

The sparse term is introduced to the problem following the same logic as RPCA algorithm [2]. In that problem the sparse part is expressed with l_1-norm, which has been a standard technique for sparse solution. Therefore, our minimization problem is reformulated as:

$$\begin{aligned} \min_{C,S,E} \quad & \frac{1}{2}||X - XCS - E||_F^2 \ + \ \lambda||E||_1 \\ \text{s.t.} \quad & C \geq 0,\ S \geq 0,\ 1^T C = 1^T,\ 1^T S = 1^T, \end{aligned} \tag{3.3}$$

After solving (3.3) matrix XCS will represent the convex hull of the low-rank part and the rest of the data, matrix E, will be the sparse part.

The problem (3.3) is not convex [22] for C, S, E simultaneously. In order to solve the problem we deploy an iterative approach that updates the values of C, S and E individually, while holding the other variables constant. Hence, a local optimal solution can be found by solving a sequence of convex optimization problems.

3.3 Optimization methodology

In order to solve the minimization problem 3.3, the associate Lagrangian function is expressed as:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}||X - XCS - E||_F^2 \ + \ \lambda||E||_1 \\ &= \frac{1}{2}\,\text{tr}((X - XCS - E)^T(X - XCS - E)) \ + \ \text{tr}(\Phi C^T) \ + \text{tr}(\Psi S^T) \\ &= \frac{1}{2}\,\text{tr}(X^T X - X^T XCS - X^T E - S^T C^T X^T X + S^T C^T X^T XCS \\ &\quad + S^T C^T X^T E - E^T X + E^T XCS + E^T E) \ + \ \text{tr}(\Phi C^T) \ + \text{tr}(\Psi S^T) \end{aligned} \tag{3.4}$$

where Φ and Ψ the Lagrangian multipliers corresponding to the non-negativity constraints.

Update C, S for fixed E

By taking the partial derivative of the Lagrangian with respect to C we have:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial C} &= \frac{1}{2}\frac{\partial}{\partial C}(\text{tr}(-X^TXCS - S^TC^TX^TX + S^TC^TX^TXCS \\ &\quad + S^TC^TX^TE + E^TXCS) + \text{tr}(\Phi C^T)) \\ &= -X^TXS^T + X^TXCSS^T + X^TES^T + \Phi,\end{aligned} \tag{3.5}$$

and by differentiating with respect to S we get:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial S} &= \frac{1}{2}\frac{\partial}{\partial C}(\text{tr}(-X^TXCS - S^TC^TX^TX + S^TC^TX^TXCS \\ &\quad + S^TC^TX^TE + E^TXCS) + \text{tr}(\Psi S^T)) \\ &= -C^TX^TX + C^TX^TXCS + C^TX^TE + \Psi\end{aligned} \tag{3.6}$$

From the Karush-Kuhn-Tucker (KKT) conditions [22] we get:

- $\phi_{ij}c_{ij} = 0$
- $\psi_{ij}s_{ij} = 0$
- $(-X^TXS^T + X^TXCSS^T + X^TES^T)c_{ij} = 0$
- $(-C^TX^TX + C^TX^TXCS + C^TX^TE)s_{ij} = 0$

From the last two equations we construct the multiplicative update rules with respect to C and S respectively:

$$c_{ij}[t] \leftarrow c_{ij}[t-1]\frac{(X^TXS^T)_{ij}}{(X^TXCSS^T + X^TES^T)_{ij}} \tag{3.7}$$

$$s_{ij}[t] \leftarrow s_{ij}[t-1]\frac{(C^TX^TX)_{ij}}{(C^TX^TXCS + C^TX^TE)ij} \tag{3.8}$$

Update E for fixed C,S

For the optimization of the sparse matrix E a different method is followed. It is known from the literature that the kind of problems having this form: $||\quad||_F + ||\quad||_1$ are convex optimization problems and in order to solve them with respect to the l_1-norm we use $Proximal\ Operators$ [16] and specifically soft thresholding or shrinkage operator. Since C, S are fixed, the optimization problem is formed as:

$$\min_E \quad \frac{1}{2}||X - XCS - E||_F^2 \ + \ \lambda||E||_1. \tag{3.9}$$

Theorem 1. *$Soft\ Thresholding\ Operator$ is defined as:*

$$\mathcal{S}_\tau[x] = \begin{cases} x - \tau,\ if\ x > \tau \\ x + \tau,\ if\ x < \tau \\ 0,\ otherwise \end{cases} \tag{3.10}$$

where $x \in \mathbb{R}$ and $\tau > 0$.

This operator can be extended to matrices by applying it element-wise. Now, consider the following l_1-minimization problem:

$$\min_{\tau} \quad \frac{1}{2}||x - \tau||_F^2 \ + \ t||\tau||_1. \tag{3.11}$$

The unique solution $\tau*$ of Equation 3.11 is given by $\mathcal{S}_\tau[x]$.

According to the Theorem 1, the optimal solution to problem 3.9 is $\mathcal{S}_\lambda(X - XCS)$. Hence, the update rule for E is:

$$E \ \leftarrow \ \mathcal{S}_\lambda(X - XCS) \tag{3.12}$$

3.4 Stopping Criteria

The algorithm below describes the whole procedure of solving this convex optimization problem that combines Archetypal Analysis for low-rank representation and proximal operators for finding the sparse part. The reconstruction error is defined as:

$$err = \frac{||X - XCS - E||_F}{||X||_F} \tag{3.13}$$

After plotting the reconstruction error for a bunch of different test sets and for maximum iterations $max\_iter = 700$, we noticed that the error plot was converging before it reaches the maximum iterations. This observation means that our algorithm tends to a solution and the optimization procedure is working. Hence, our first stopping criterion is activated when the remainder between the current and previous iteration's reconstruction error tend to zero with tolerance $tol = 1 \times 10^{-3}$.

Another convergence criterion has to do with the optimization success of each matrix individually [18]. For this criterion we create a condition that says: when the maximum value of the three individual errors of the matrices C,S and E is smaller than a threshold then we have guarantees about the convergence.

We decided the threshold value empirically. Having decided the maximum of iterations, as we mentioned before, the error for each matrix is saved in a vector and plotted, in order to observe how it decreases as the iterations increase. The goal is to decrease until it is almost stabilized. We conducted this experiment for a test set of 10% of the songs and observed that our goal is achieved.

The threshold value we chose is 1×10^{-3}. So the stopping criterion is formulated as:

$$A = \frac{||C^t - C^{t-1}||_F}{||X||_F},\ B = \frac{||S^t - S^{t-1}||_F}{||X||_F}, C = \frac{||E^t - E^{t-1}||_F}{||X||_F}$$

and the algorithmic procedure we followed is:

$Converged \leftarrow False,\ iter = 0,\ e = 1 \times 10^{-3}$
if $maximum(A, B, C) < e$ **then**
    $Converged = True$
else
    $Continue\ until\ iter\ =\ 700$
end if

Thus, if both of the stopping criteria are activated or the algorithm reaches the maximum of iterations then the iteration procedure stops.
Finally, we summarize our Robust Archetypal Analysis algorithm in Algorithm 1.

3.5 Algorithm

Algorithm 1 Archetypal Analysis with Sparsity Constraints

Require: $Data : \{X \in \mathbb{R}^{m \times n}\}. Parameters :\ \lambda, k, tol$

Initialization: Initialize matrices C,S with random values and E with zero values
Converged = False
while Converged = False **do**
    $C[t] \leftarrow C[t-1](X^T X S^T)/(X^T X C S S^T + X^T E S^T)$
    Normalize columns of $C[t+1]$ to unit sum
    $S[t] \leftarrow S[t-1](C^T X^T X)/(C^T X^T X C S + C^T X^T E)$
    Normalize columns of $S[t+1]$ to unit sum
    $E[t]\ \leftarrow\ \mathcal{S}_\lambda(X - XCS)$
    Check stopping criteria:
    if stop_crit_1 and stop_crit_2 or iter=max_iter **then**
        Converged = True
    end if
end while

4. EXPERIMENTAL EVALUATION

In this chapter we will analyse the experimental procedure we followed and the results we obtained. Our main goal was to build a framework that will have its roots in the logic of Singing Voice Separation using Robust Principal Component Analysis(RPCA) [2], presented analytically in Chapter 2.2.3, and optimize it using Archetypal Analysis Method. To achieve this we firstly reproduced RPCA method using python language in order to compare these results with our experiment's.

For the performance evaluation of the proposed method we used MIR-1K dataset[1] and Blind Audio Source Separation(BASS) evaluation metrics [19]. Finally we got the desired results and compared them with RPCA's. In the rest of the chapter we analytically present our data, the evaluation metrics we used, the parameter selection procedure and the results of the comparison.

4.1 MIR-1K Dataset

For the evaluation procedure we used MIR-1K dataset [5] that consists of 1000 song clips recorded at 16 kHz samplerate, with duration from 4 to 13 seconds. These clips are extracted from 110 karaoke Chinese pop songs that are sung by both male and female amateur singers. The music accompaniment and the vocals are recorded separately in left and right channels respectively. This tactic offers the advantage that anyone who uses the dataset has the real sources available in order to evaluate the performance. MIR-1K dataset also contains manual annotations of pitch contours, unvoiced frames, indices of the vocal and non-vocal frames and lyrics.

To proceed to source separation we need our sources to be monaural. So the dataset is converted to mono by averaging the left and right channels.

4.2 Audio Representation

Our data, coming from MIR-1K dataset, are 1000 wav clips. We have to represent these files in a form suitable to perform separation. By performing the Fourier Transform in an audio track we get its most unprocessed form, the waveform, which is a time domain representation of a signal. Some source separation approaches operate on the waveform directly, although many require some prepossessing before separating sources.

Waveforms are a time-domain representation of signal and by applying the Discrete Fourier Transform(DFT) to them we get the frequency representation[2]. It would be more useful to know when each frequency is present, so we need a time-frequency representation i.e. spectrogram. To obtain the matrix that we will use for the separation procedure and its representation we perform Short Time Fourier Transform(STFT)(check Appendix A) using a hann-window of size of 1024 and a hop size of 256, at samplerate $sr = 16000$. The magnitude of the spectrogram we obtained of the STFT constitutes the input to Archetypal Analysis algorithm.

[1] https://sites.google.com/site/unvoicedsoundseparation/mir-1k

[2] The presence of each frequency component of the signal across its whole duration

4.3 Audio Reconstruction

The results we obtained after Archetypal Analysis algorithm (Algorithm 1) are two 2-Dimensional matrices that correspond to two magnitude spectrograms. One for the low-rank part and one for the singing voice part.

In order to turn the magnitude spectrogram of their estimated source back into a waveform so that someone may listen to the source estimation, the magnitude spectrogram has to be inverted to its complex form i.e restoration of the phase component. A good strategy that is usually preferred for this kind of problems [2] is to copy the phase from the mixture. Let a mixture STFT $Y \in \mathbb{C}$:

$$\tilde{X}_i = \hat{X}_i \odot e^{j \cdot \angle Y}$$

where $\hat{X}_i \in \mathbb{R}$ represents a magnitude spectrogram of the source estimate, $j = \sqrt{-1}$, "$\angle$" represents the angle of the complex-valued STFT, $\tilde{X}_i \in \mathbb{C}$ indicates that the estimate for source i is complex-valued similar to an STFT and $\odot$ denotes the Hadamard product(element-wise product). After that, we calculate inverse Short-time Fourier Transform(ISTFT) to covert the signal back to waveform.

4.4 Evaluation Metrics

In this section we will describe the most popular evaluation metrics that are used as objective measures for Blind Audio Source Separation(BASS) problems. Firstly we will present BASS evaluation metrics, and then a global performance measurement named GNSDR.

4.4.1 SDR,SIR,SAR

In [19] the authors proposed a framework for evaluation of BASS problems' results. One of the assumptions they made is that the true source signals are known and that the mixing system or the demixing technique don't have to be known (the other assumptions refer to noise and allowed distortions that are out of our scope). In our experiment all two true sources are known.

Firstly, the estimate source signal $\hat{x}_i$ is decomposed in the target source plus three error terms:

$$\hat{x_i} = x_{target} + e_{interf} + e_{noise} + e_{artif} \tag{4.1}$$

- x_{target} is the signal we target to find
- e_{interf} is the estimated component of other sources interference to the signal
- e_{noise} is the estimated noise component
- e_{artif} is the estimated component of unwanted artifacts (resulting from the editing or manipulation of a sound)

These four terms represent the sound we perceive: $\hat{x}_i$, that comes form the wanted source x_i, other unwanted sources $x_{i'}$, sensor noises n_j and other distortions.

The performance criteria come from three orthogonal projections of signals. Let $\Pi\{x_1, .., x_I\}$ be the orthogonal projector in the subspace spanned by vectors $x_1, x_2, ..., x_n$, such that:

$$P_{x_i} = \Pi\{x_i\} \tag{4.2}$$

$$P_{\mathbf{x}} = \Pi\{(x_{i'})_{\forall 1 \leq i' \leq n, i \neq i'}\} \tag{4.3}$$

$$P_{\mathbf{x,n}} = \Pi\{(x_{i'})_{\forall 1 \leq i' \leq n}, (n_j)_{\forall 1 \leq j \leq m}\} \tag{4.4}$$

The estimate source signal $\hat{x}_i$ is the sum of the following terms:

$$x_{target} = P_{x_i}\hat{x}_i \tag{4.5}$$

$$e_{interf} = P_{\mathbf{x}}\hat{x}_i - P_{x_j}\hat{x}_i \tag{4.6}$$

$$e_{noise} = P_{\mathbf{x,n}}\hat{x}_i - P_{\mathbf{x}}\hat{x}_i \tag{4.7}$$

$$e_{artif} = \hat{x}_i - P_{\mathbf{x,n}}\hat{x}_i \tag{4.8}$$

Finally the calculation of performance measurements of the similarity between $\hat{x}_i$ and x_i expressed in decibels (dB) are:

Source-to- Artifacts Ratio

$$\text{SAR}_{x_{target}} = 10log_{10}\frac{||x_{target} + e_{interf} + e_{noise}||^2}{||e_{artif}||^2} \tag{4.9}$$

This is usually interpreted as the amount of unwanted artifacts a source estimate has with relation to the true source.

Source-to-Interference Ratio

$$\text{SIR}_{x_{target}} = 10log_{10}\frac{||x_{target}||^2}{||e_{interf}||^2} \tag{4.10}$$

This is usually interpreted as the amount of other sources that can be heard in a source estimate. This is most close to the concept of “bleed”, or “leakage”.

Source-to-Distortion Ratio

$$\text{SDR}_{x_{target}} = 10log_{10}\frac{||x_{target}||^2}{||e_{interf} + e_{noise} + e_{artif}||^2} \tag{4.11}$$

SDR is usually considered to be an overall measure of how good a source sounds. If a paper only reports one number for estimated quality, it is usually SDR.

4.4.2 Global normalized source-to-distortion ratio (GNSDR)

In order to have an overall view of our separation performance we need a metric to compare the improvement of the distortion between the original source and the estimated one. This metric is called Normalised Signal to Distortion Ratio(NSDR), which measures the difference of the SDR in decibels(dB) between the original mix and the estimated source. In the context of audio source separation and in order to perform comparison with other methods we are mainly interested in voice estimation. Therefore, the separation performance is evaluated using the voice NSDR and not the music one.

$$NSDR(\hat{x}, v, x) = SDR(\hat{v}, v) - SDR(x, v)$$

where $\hat{x}$ and x are the estimated and the original mix input source respectively, $\hat{v}$ is the estimated vocals and v is the original voice. For overall separation performance, the Global NSDR (GNSDR) was calculated by taking the mean of the NSDRs over all the mixtures of each set, weighted by their length. Higher values of GNSDR mean better separation.

$$GNSDR(\hat{X}, V, X) = \frac{\sum(NSDR(\hat{X}, V, X) \times X_{duration})}{\sum X_{duration}}$$

4.5 Experimental Results

In this section we present the details of our experimental procedure and analyze the results we obtained. Our goal is to estimate the separation quality of our model and especially the singing voice part.

We implemented our experiment in Python from scratch in order to have a base to compare it with RPCA code that was originally implemented in MATLAB[3]. For matrix operations we made use of numpy and scipy libraries and matplotlib library for visualization. After the successful algorithmic to code translation we experimented mostly with the initialization of matrices and the hyperparameter tuning. The proposed method includes two hyperparameters that have to be optimal to improve the quality of our results: the number of archetypes k and the trade-off parameter λ.

4.5.1 The effect of λ

Parameter λ is the trade-off parameter between the sparsity of matrix E, that represents the vocal part, and the rank of low-rank matrix XCS, that represents the background music part. Matrix E is sparser when λ has higher values and vice versa. We conducted many experiments with different values of λ and noticed that the selection of this parameter is crucial for the separation quality. We chose the value range empirically ending up to these five values {0.1,0.5,1.0,2.0,3.0}.

[3]https://github.com/posenhuang/singingvoiceseparationrpca

The experimentation showed that in higher values of λ, which is translated in sparser matrix E, the interference is reduced unlike artifacts that are increased and opposite. When a matrix is sparse some parts of the signal are deleted so artifacts are created. So if we reduce the sparsity of E we will also reduce the artifact error, so the Signal to Artifacts Ratio (SAR) is increased. In figure (6) we can observe the aforementioned.

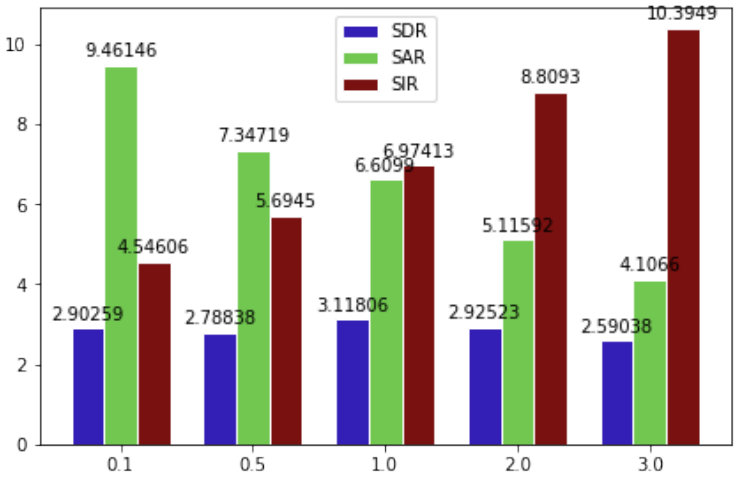


Figure 6: Grouped barplot chart for vocals, for different values of λ

Table 1: Results for vocals for different λ values

	SDR	SIR	SAR
λ=0.1	2.90259	9.46146	4.54606
λ=0.5	2.78838	7.34719	5.6945
λ=1.0	3.11806	6.6099	6.97413
λ=2.0	2.92523	5.11592	8.8093
λ=3.0	2.59038	4.16066	10.3949

Also, we can observe that Signal to Distortion Ratio (SDR) values range from 2.5 to 3.1, ending up to choose λ=1.0 which achieves the higher value. This value is translated to good separation results when unsupervised machine learning methods are used for singing voice separation. Also SAR, SIR present a good behaviour, but SDR is the basic evaluation metric from which we chose the best version for our model.

4.5.2 The effect of k

Choosing the number of archetypes is a very important task because it defines the quality of the entire separation procedure. Matrix XCS is low-rank so to make sure that we have a number of archetypes that corresponds to the background music of each song we took the rank of the separated low-rank matrix from Robust PCA method and we set the number of archetypes k equal to this rank.

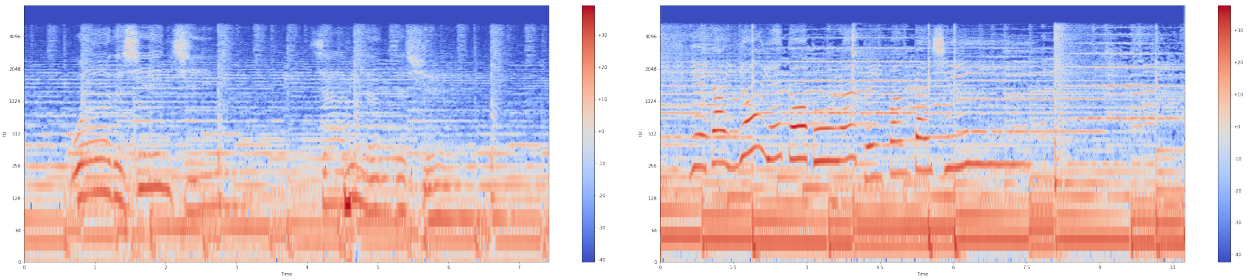
4.5.3 Comparison with Robust Principal Component Analysis

The main point of our experiment is to prove that the quality of singing voice separation improves if the music accompaniment part is bounded in a more clear way, i.e. by introducing archetypes to the separation method. To prove this statement we compare our experimental results with Huang et al.'s [2] experiment, that performs Singing Voice Separation (SVS) using Robust Principal Component Analysis (RPCA). To achieve this we reproduced Huang et al.'s [2] experiment in Python code so that we can compare the results with our model.

We chose two characteristic, well separated wav files that consist of a man's voice (stool_4_07.wav) and a woman's voice (amy_7_08.wav) to highlight the improvement of the separation through our method. In figure(7) we see the visualization[4] of Log Power Spectrograms[5] that are produced from the mixed signal for these two wav files.

In figures (8) and (10) we can observe the visualization of the separated background music part using RPCA and Archetypal Analysis method. Music accompaniment seems to be better separated when the separation method used is Archetypal Analysis. As we can see in the heatmaps, darker colours, that indicate higher amplitudes, are gathered in low frequencies and they have repetitive schema in both pictures. In Archetypal Analysis depiction, though, they have almost no interference from the vocals (sparse higher rank dark colours or spikes) like RPCA's spectrogram has. This last observation makes Archetypal Analysis a better separation method.

This confirms our claim that when the music accompaniment part is improved then the quality of the vocal part is also improved. As we can observe in the representations in figures 9 and 11, the singing voice is clearly better separated when separation method used is Archetypal Analysis. Likewise the background case, vocals in the Archetypal Analysis magnitude spectrogram have almost no interference from the background part.

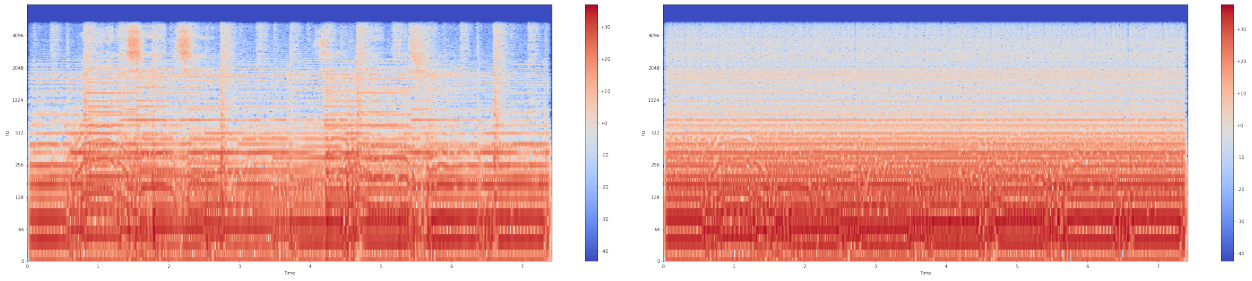


(a) stool_4_07.wav **(b) amy_7_08.wav**

Figure 7: Spectrogram visualization for source mix signal

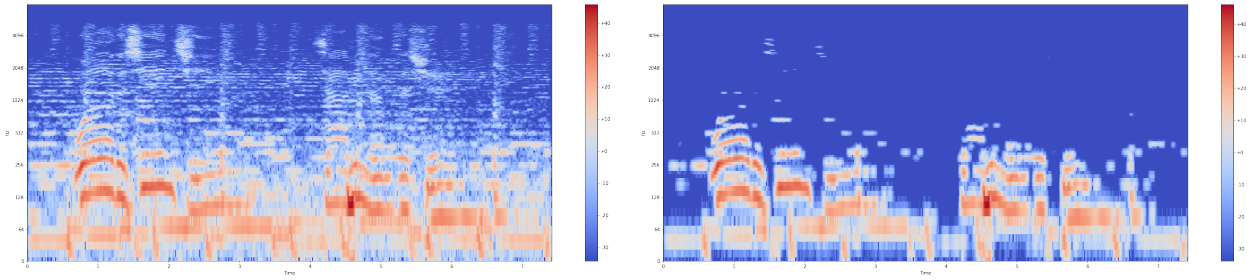
[4]Each TF bin in the heatmap represents the amplitude of the signal at that particular time and frequency. The brighter colors indicate high amplitudes and darker colors low amplitudes.

[5]For a complex-valued STFT, $X \in \mathbb{C}^{T \times F}$, the Log Spectrogram is calculated taking the log of the square of each element in the STFT, $\log |X|^2 \in \mathbb{R}^{T \times F}$.



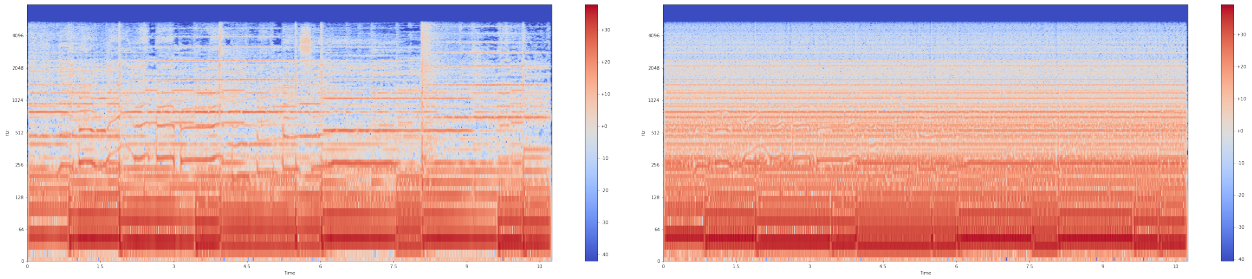
(a) RPCA (b) Archetypal Analysis

Figure 8: Separated background music for stool_4_07.wav



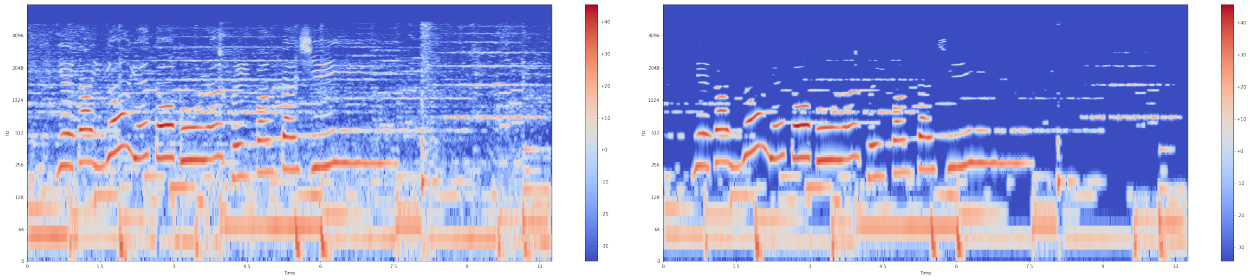
(a) RPCA (b) Archetypal Analysis

Figure 9: Separated singing voice for stool_4_07.wav



(a) RPCA (b) Archetypal Analysis

Figure 10: Separated background music for amy_7_08.wav



(a) RPCA (b) Archetypal Analysis

Figure 11: Separated singing voice for amy_7_08.wav

Another usefull representation is the one that provides the isolated components of each method. Archetypal Analysis is based on the finding of the correct number of archetypes that better describe the background part. As we already mentioned, the number of archetypes is smaller than the number of samples or the frequency bins of the STFT. In the figures below we compare three out of eight archetypes taken from the separated background music of ”yifen_2_11.wav”, with the three first principal components that came out of RPCA separation method.

In figure (12a) we see the rows of the matrix XC, that contain the archetypes and in figure(12b) are the columns of the matrix S that contain the activations of the archetypes i.e their weights.

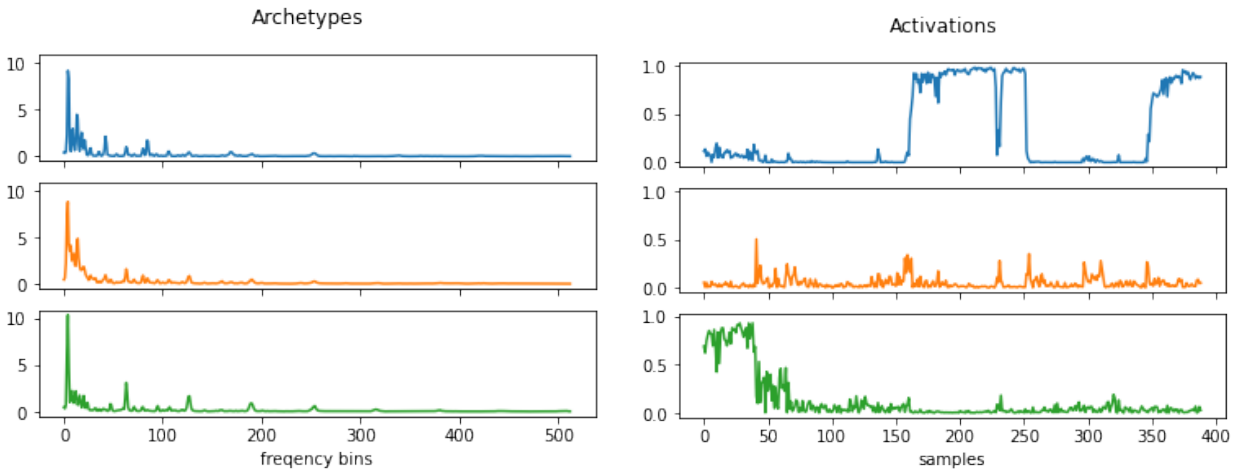


Figure 12: Archetypal Analysis, yifen_2_11.wav

For the RPCA method we computed the singular value decomposition(SVD) of the background music magnitude spectrogram for the largest k=3 singular values. This computation gave us two matrices, W and H. The rows of W (figure 13a) represent the most dominant principal components of the background music, or the eigenvectors that correspond to the frequencies, and the columns of H (figure 13b) represent their activations in the time domain, or their weights that vary over time.

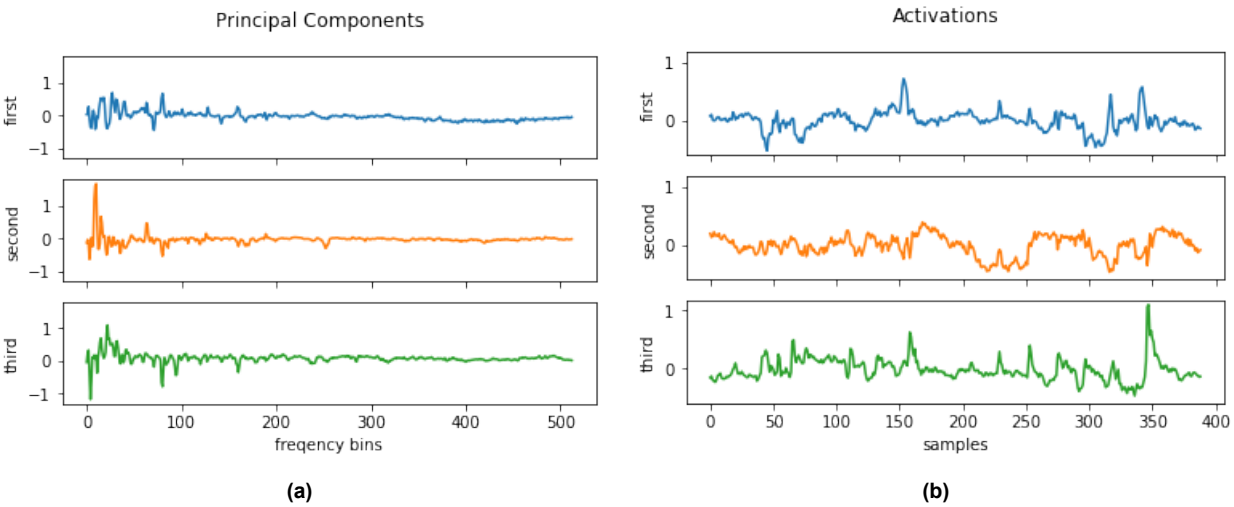


Figure 13: Robust Principal Component Analysis, yifen_2_11.wav

From the above visualization we can notice that the archetypal analysis method gives us a better representation of the frequency patterns that exist in a song. All eight archetypes of ”yifen_2_11.wav” provide a full description of the background music part, and specifically the rows of XC contain its spectra while the columns of S contain the respective temporal activity.

The representation above is useful because it is easier for us to confirm that the background music is low rank by observing the frequency patterns that exist in each archetype. The thing that makes the archetype representation better is the non-negativity factor that is the result of the archetypal analysis algorithm that takes advantage of the spectrogram’s non-negative nature. So the plots contain only non-negative values and it is easy for us to get information about archetypes behaviour.

Finally we compare these two methods in terms of Global Normalized Signal to Distortion Ratio (GNSDR) which is a globalized metric that authors of ”SVS using RPCA” [2] used to present an overall picture of the separation quality. We conducted the experiment in the 70% of the MIR-1K dataset for the best version of both models and we obtained the following metrics:

Table 2: Comparison between Huang [2] and our approach in terms of GNSDR.

Model	RPCA for λ=1.0	Archetypal Analysis for λ=1.0
GNSDR	3.39dB	3.13dB

Table 2 gives us the final guarantee that our model works as good as Huang et al.’s [2].

5. CONCLUSION AND FUTURE WORK

In this thesis, we proposed an unsupervised approach which applies archetypal analysis with sparsity constraints on singing-voice separation from music accompaniment for monaural recordings. Our methodology was based on the assumption that the background music lies on a low rank subspace, while singing voice is sparse within the song. We enriched this assumption by introducing the archetype factor, which implies that the background music can be described better by certain number of archetypes according to its inherent characteristics. This innovation improved the separation quality of the music accompaniment and as a result the quality of the separated singing voice. Hence, we achieved to build a strong base for better separation results in the future.

We examined the parameter λ_k and the parameter k (the number of archetypes) and tuned them to improve our model's performance. We also performed an overall comparison with RPCA method using the same dataset and the same evaluation metrics. We managed to achieve almost the same separation quality with SVS using RPCA, indicated by the global metric (GNSDR) that reached values bigger than 3.0dB.

The visualization of our results in the previous chapter gives us a detailed view of the separated results. This is a useful tool for us to analyze a song's information in order to improve our method in the future and it is also an indicator that if we perform some more optimization techniques, the proposed model will surpass RPCA's results.

In future work we can expand our experiments by investigating dynamic parameter selection methods according to different contexts, try different Signal to Noise Ratio (SNR) [19], that control the energy levels of both singing voice or music accompaniment and also perform ideal binary mask to our STFT matrix to check if it improves the separation results.

Another interesting experimentation can be the estimation of phase component. In our method we copied the phase from the initial mix (Chapter 4.3), which is an easy and fast way to return our spectrogram back to its complex form in order perform ISTFT and listen to the song, but this strategy has a certain drawback. If we use the phase from the source mix signal and multiply it with the separated singing voice signal, the vocal part will contain mix's phase and as a result the output will contain artifacts.

Finally, another direction that has guaranteed good separation results is around deep learning methods. But these methods are supervised and surpass the scope of our thesis which is singing voice separation without the use of prior knowledge i.e. unsupervised.

ABBREVIATIONS - ACRONYMS

SVS	Singing Voice Separation
BSS	Blind Source Separation
STFT	Short Time Fourier Transform
RPCA	Robust Principal Component Analysis
ICA	Independent Component Analysis
NMF	Nonegative MATRIX Factorization

APPENDIX A. DEFINITIONS

Definition A.1 (Short Time Fourier Transform)

Short Time Fourier Transform (STFT) is a time-frequency (TF) representation of a signal. An STFT is calculated from a waveform representation by computing a discrete Fourier transform (DFT) of a small, moving window across the duration of the window. The location of each entry in an STFT determines its time (x-axis) and frequency (y-axis). The absolute value of a TF bin $|X(t, f)|$ at time t and frequency f determines the amount of energy heard from frequency f at time t.

Each bin in STFT is complex, meaning each entry contains both a magnitude component and a phase component. Both components are needed to convert an STFT matrix back to a waveform so that we may hear it.

Complex valued STFT is invertable, i.e. it can be converted back to waveform. This transform is called inverse-STFT (ISTFT)

STFT has two parameters that have to be defined, window length and hop length. The window length determines how many samples are included in each short-time window. Due to how the DFT is computed, this parameter also determines the resolution of the frequency axis of the STFT. The longer the window, the higher the frequency resolution and vice versa. The hop length determines the distance, in samples, between any two adjacent short-time windows.

Definition A.2 (Frobenius Norm)

The Frobenius norm is matrix norm of an $m \times n$ matrix A defined as the square root of the sum of the absolute squares of its elements [23]:

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{i=1}^{n} |a_{ij}|^2}$$

REFERENCES

[1] Jonathon Shlens. A tutorial on independent component analysis. *arXiv preprint arXiv:1404.2986*, 2014.

[2] Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60. IEEE, 2012.

[3] Christian Bauckhage. Numpy/scipy recipes for data science: Archetypal analysis via frank-wolfe optimization. *researchgate. net, Oct*, 2020.

[4] Yipeng Li and DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1475–1487, 2007.

[5] Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE transactions on audio, speech, and language processing*, 18(2):310–319, 2009.

[6] Alexey Ozerov, Pierrick Philippe, Frdric Bimbot, and Rmi Gribonval. Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1564–1578, 2007.

[7] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, pages 477–482, 2014.

[8] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[9] D Seung and L Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[10] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[11] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.

[12] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684)*, pages 177–180. IEEE, 2003.

[13] Guy J Brown and Martin Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994.

[14] J-L Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298, 1994.

[15] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[16] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[17] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[18] Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *Advances in neural information processing systems*, 24, 2011.

[19] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[20] Zafar Rafii and Bryan Pardo. A simple music/voice separation method based on the extraction of the repeating musical structure. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 221–224. IEEE, 2011.

[21] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

[22] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[23] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.