Blind Source Separation for Automatic Music Transcription

Capstone Report

Bauyrzhan Kurmangaliyev

Nazarbayev University
Department of Electrical and Computer Engineering
School of Engineering and Digital Sciences

Copyright © Nazabayev University

This project report was created on TexStudio editing platform using LaTeX. All the figures were drawn using draw.io online software tool.

NAZARBAYEV UNIVERSITY

Electrical and Computer Engineering
Nazarbayev University
http://www.nu.edu.kz

Title:
Blind Source Separation for Automatic Music Transcription

Theme:
Latent Dimensionality Estimation for Nonnegative Matrix Factorization

Project Period:
Fall 2023 Spring 2024

Project Group:
Applications of Signal Processing Laboratory (ASP-LAB)

Participant(s):
Bauyrzhan Kurmangaliyev

Supervisor(s):
Muhammad Tahir Akhtar

Copies: 1

Page Numbers: 48

Date of Completion:
June 2, 2024

Abstract:

The primary objective of this project is to develop methods aimed to the conduct the blind signal separation of musical notes with Nonnegative Matrix Factorization (NMF). This is motivated by the fact that music signals are often recorded with a single microphone, hence, there is a need to develop the Automatic Music Transcription (AMT) methods that could mitigate this assumption and produce the desirable separation result. Therefore, this project report presents the rank estimation method for determination of number of musical notes in the recording. It is motivated by the fact that most of the research works on NMF assume *a priori* knowledge regarding the rank of factorization which may not be available in most of the real world scenarios. As a result, the Weighted Singular Value Thresholding based on Stein's Unbiased Risk Estimate (WSVT-SURE) in which rank estimation is performed by non-uniform shrinkage of singular values via weight vector is presented. We also introduce gradient optimization of a smooth approximation of WSVT-SURE (GWSVT-SURE) to estimate the optimal threshold parameter. In the context of AMT, the proposed algorithms allow one to estimate the number of musical note components in the recordings. The proposed algorithms have been evaluated with the polyphonic piano music excerpts. It is observed that the proposed WSVT-SURE algorithm reaches significant improvement in the estimation performance, while GWSVT-SURE shows substantial savings in the computational cost.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author(s).

Contents

Preface

Music is a universal language that unites cultures and people with different backgrounds. It is a source of fascination and inspiration that is closely tied with human history from the early ages. This project is a synergy between technology and art which demonstrates the analysis and transcription of musical expression. Specifically, the area of interest concerns many real-world situations where diverse music sounds emanate from multiple instruments and mixed up together, the particular problem revolves around recovering the desired musical excerpt from the complex mixture. The outcome of this project will open new possibilities in music education, enabling students to assess the precision of their musical proficiency. Additionally, it offers musicians the ability to recreate and reinterpret classical music compositions. Development of this project might also be beneficial for music streaming companies in the sense that it allows them to develop better music recommendation and genre identification algorithms. In future perspectives, developed AMT systems may be implemented in portable devices and used for the analysis and correction of musical data.

Chapter 1 focuses on the **Introduction** of Blind Source Separation which is the process of separating a latent source signals from the observable mixtures. However, more realistic case restricts the number of sensors, and assumes it to be one, which left us with only single mixture. Although, it is very difficult to conduct any separation, algorithms such as Nonnegative Matrix Factorization (NMF) is regarded as the best tool for solving this problem. NMF paradigm for source separation concerns factorizing mixture spectrogram into frequency and time information matrices. This report focuses on specific problem of latent dimensionality estimation, in other word, estimating the rank of NMF. NMF rank in the applications of music processing could represent the number of musical notes, instruments or singers.

Chapter 2 gives details of **Literature Review and Related Work** including fundamental algorithms of NMF, by considering different variations and providing mathematical derivations. They include basic NMF with Multiplicative Update (MU) based optimization considered for various statistical measure (cost functions). Another part concerns the alternating optimization scheme, which speeds

the NMF algorithm allowing it to be used with large data. Also, the big part of this chapter includes preliminary study for Singular Value Thresholding (SVT) and Stein's Unbiased Risk Estimate (SURE) which is required to understand for the development and design of the proposed algorithms.

Chapter 3 presents the **Proposed Algorithms** on weighted SVT-SURE and the gradient optimization of its risk function. The gradient based optimization required to approximate the risk via substituting non differentiable functions with their smooth alternatives.

Chapter 4 provides **Experimental Results** on implemented NMF algorithms for convergence and computational complexity together with the evaluation of the proposed algorithms. Proposed algorithms are evaluated for musical note estimation performance with two metrics, namely Mean Absolute Deviation (MAE) and Mean Absolute Deviation (MAD). MAE measures the average error in estimation, whereas MAD computes the deviation of the estimation from the MAE. Also, proposed algorithms are evaluated for computational complexity by measuring the elapsed time for different dimensionalities of the data.

Chapter 5 provides **Conclusion** with summary for the work done in the study. The proposed algorithm reaches vast improvements in note estimation accuracy and efficiency compared with benchmark algorithms. The chapter also presents the ideas for the future work that would eliminate the dependence of the proposed algorithm on the adjustments of weight parameter.

Nazarbayev University, June 2, 2024

Bauyrzhan Kurmangaliyev
<bauyrzhan.kurmangaliyev@nu.edu.kz>

Acknowledgments

I want to express my deepest gratitude to my family: Saulesh, Kairat, Yerassyl, and Daniya. Your love, care, and unwavering support from the earliest stages of my life have been the foundation of my academic journey. It is through your encouragement and belief in me that this work has come to light and I am infinitely grateful to be a part of such a wonderful family. Your influence will be evident in every of my future work I undertake.

I am also infinitely grateful to my supervisor, Professor Muhammad Tahir Akhtar, for his guidance and support throughout the process of developing this project. I want to thank my Professor for organizing group research seminars and lectures to cover theoretical aspects of my topic. His dedication to teaching and expertise have a profound impact on my growth as a researcher.

I also want to extend my appreciation to my friends: Dimash, Diyar and Zhantlek for their moral support and countless moments of joy and laughter. Indeed, they have been the greatest source of strength and immense happiness throughout all my years of study. I sincerely want our mutual support and companionship continue till eternity.

Furthermore, I want to take a moment and appreciate the Applications of Signal Processing Laboratory (ASP-LAB) including Professor Muhammad Tahir Akhtar and lab members. The support and invaluable resources have been crucial throughout the duration of this research. A collaborative environment, mentorship and encouragement provided by ASP-LAB has greatly enriched my academic journey.

This Capstone project would have been impossible without all these individuals, and I deeply appreciate their presence in my life.

...Amicitiae nostrae memoriam spero sempiternam fore

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Automatic Music Transcription (AMT) aims to extract meaningful information from the music signals and represent them by means of Musical Instrument Digital Interface (MIDI) or other conventional standards. This is executed mainly by developing computational algorithms for specific problems. AMT comprises multiple subtasks which include pitch [1], onset, offset [2] detection, beat tracking [3], chord estimation [4], music segmentation [5], musical instrument [6], note [7] recognition and genre classification [8]. Figure 1.1 summarizes the key research directions of AMT and their corresponding sub-tasks in the form of taxonomy. Overall, due to the wide range of sub-task, AMT is considered to be a fundamental problem in the applications of Music Information Retrieval (MIR) [9]. It is a challenging signal processing problem which gets complicated by the fact that the signal under consideration might be the polyphonic mixture of musical notes. It results in overlapping notes played from different instruments that interfere in time and frequency thus making music data complicated to process. The rich complexity and diverse range of musical data makes polyphonic transcription an unsolved problem in the realm of AMT [10].

In this situation where unobserved acoustic signals are mixed in an unknown environment, the process of Blind Source Separation (BSS) is applied towards recovering the source signals from the mixture of sources. The term *Blind* represents the scenario where we do not possess any information both on source signals and the mixing process. This feature may seem to be natural for humans, but exceptionally difficult for machines. Although the conventional BSS algorithms such as Independent Component Analysis (ICA) [11] have successfully applied for this task, the primary assumptions of ICA restricts it to be applied in realistic cases. The assumptions include:

1. Statistical independence and non-Gaussian distribution of source signals.

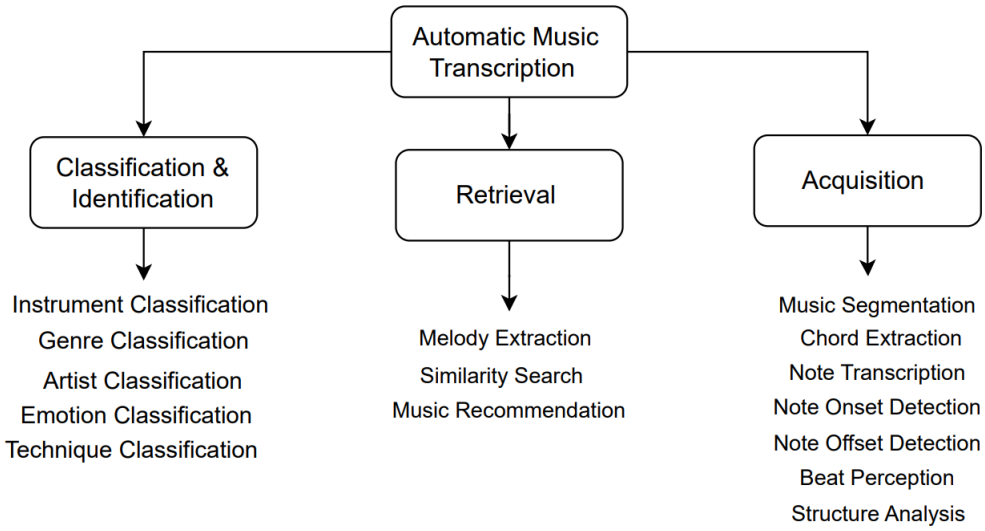

Figure 1.1: The sub-tasks of Automatic Music Transcription

2. Linear mixing process of the source signals via mixing matrix.
3. The order of the estimated sources match input source signals up to permutation matrix.
4. The match between number of source signals and sensors.

The latter assumption does not meet most of the real-life scenarios where a mixture signal is often observed only from the one sensor (single channel). For example, when performing sound recordings we are often limited with the number of recording equipment. Also, in everyday life, when someone needs to records the acoustic signals, there are very few number of microphones embedded in the smartphones. In fact, this variation of BSS is known as Single-Channel Blind Source Separation (SCBSS). SCBSS is aimed to find the latent sources given only a single observed mixture, and it is an extreme case of underdetermined BSS where the number of sensors (microphones) is less than the number of sources:

$$x(n) = \sum_{i=1}^{N} a_i s_i(n), \tag{1.1}$$

where N is the number of input source signals, $s_i(n)$ is the i^{th} source signal, a_i represent the mixing coefficient and $x(n)$ is the observed signal. Figure 1.2 displays the pipeline of BSS, in which the separation is conducted via the inverse of mixing matrix. In the single channel case, there are several challenges compared to conventional BSS. One of the main complexities includes deficiency of available data,

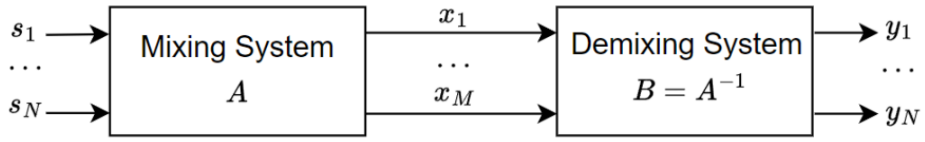

Figure 1.2: The pipeline of Blind Source Separation problem

in other words there is not enough information available on the relation between different recorded data. Therefore, applying information theoretic or statistical measures may not give decent separation quality [13]. Furthermore, referring to the Figure 1.2, the mixing matrix is not square in general, therefore even if the mixing matrix is known, there is no possibility to recover the source signals, as it does not admit the inverse. To mitigate the following complexities, Nonnegative Matrix Factorization (NMF) algorithm has been widely applied in the area of SCBSS. This is mainly because source vectors in NMF inference are modeled as a linear combination of basis vectors. Alternatively, it exploits spectrogram factorization of the single input recording into corresponding product of frequency and time factor matrices [14]. One of the utilities of matrix factorization algorithms relies on building the meaningful representation of time and frequency content of the acoustic data. Additionally, due to the well-established matrix algebra, it allows one to design advanced algorithms in the context of source separation.

1.2 Existing Approaches

NMF is an algorithm that aims to perform an approximate decomposition of a given source signal matrix representation $\boldsymbol{V}$ into a linear combination of basis vectors $\boldsymbol{W}$,$\boldsymbol{H}$ with the nonnegativity constraint of the input data.

$$\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}. \tag{1.2}$$

NMF has a variety of applications consisting of hyperspectral imaging [15], speech recognition [16], [17], and image processing [18], [19]. In AMT framework, $\boldsymbol{W}$ and $\boldsymbol{H}$ represent the frequency and activation time respectively. Nonnegativity assumption comes from the nature of signals that appear to be nonnegative, they include image pixels, magnitude spectra, and text data etc.

NMF literature includes numerous algorithms aimed to compute the meaningful decomposition of the input matrix. One approach towards it is to impose a statistical divergence measure as a cost function. For example, [20] developed a Maximization-Minimization algorithm based on beta divergence which represents a group of statistical divergence functions varying on beta parameter. These functions include Kullback-Leibler, Euclidean distance and Itakuro-Saito divergences,

the latter is widely used for music transcription tasks. On the other hand, several efficient NMF algorithms based on matrix optimization schemes have been proposed. They include Projected gradient NMF (PGD-NMF), a variant of gradient based NMF where the data matrix is further projected on the feasible set of nonnegative numbers [21], Alternating Least Squares (ALS-NMF) algorithm [22], an extension of Nonnegative Least Squares Problem (NNLS) which performs alternating minimization of a pair of update equations corresponding to factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. Additionally, Hierarchical Alternating Least Squares NMF (HALS) [23] is a rank-1 factorization approach based on updating a set of cost functions with respect to the pair of vectors of matrices. Although these methods are robust and efficient variations of NMF, AMT is mostly interested in the quality of factorization. Hence, [24] has been presented a convolutive model of NMF with additional temporal continuity constraint, this feature allows catching the frequency variations during factorization which results in qualitative separation of musical data.

Additionally, music transcription with NMF is very sensitive to the latent dimensionality of the data. Considering the NMF algorithms, latent dimensionality is equivalent to the number of basis vectors (rank) employed in factorization. It is observed that rank estimation methods which solely relies on statistical information may result in inaccurate estimation results [25]. Therefore, [26] incorporates music information in order to aid the rank estimation system. This approach is known as Computational Auditory Scene Analysis (CASA) and it is extensively employed in BSS tasks with some successful implementations including harmonic structure modeling [27], perceptually enhanced NMF [28] etc. Another attempts aimed to correctly identify the rank of factorization includes [29] which uses information theory methods based on Minimum Description Length (MDL) and assessing the encoding of the error matrix. Bayesian methods [30] with Automatic relevance determination with linkage factor between basis vector and corresponding activation vector (ARD-NMF). Nevertheless, it has been shown that [31] that rank selection methods are also dependent on the data, which means that different algorithms may perform better/worse on different datasets. Another algorithm for rank estimation has been presented in [32] which utilizes rank estimation with Stein's Unbiased Risk Estimate employing the Noisy PCA model (nPCA-SURE). The extension of aforementioned work for AMT task directed towards Rank Estimation with SURE (RESURE) and employs noise variance estimation via second largest noise eigenvalue of a music signal [33]. A similar study has applied Singular Value Thresholding (SVT) with SURE (SVT-SURE) [34] which finds the optimal thresholding parameter through minimizing the risk function, resulting in estimation of an effective number of singular values. In this situation, the threshold selection is performed by exhaustive search which is a computationally expensive procedure. Hence [35] presented a gradient optimization method based on weak differentiability of a risk estimate and approximation by finite difference method.

However, among the existing studies, there has been limited emphasis on the flexibility of an estimator. One way of possessing the control over the estimator has been presented in [36]. This work investigated a weighted variant of a SVT (WSVT) in which the threshold parameter is scaled with a weight vector producing non-uniform shrinkage. However, there is still indeterminacy on the choice of proper threshold parameter.

1.3 Problem Statement and Main Contributions

The main objective of this study is to develop accurate and efficient algorithm for musical note estimation of a piano recording. This is mainly motivated by the fact that before proceeding to the development of NMF-based AMT system, determination of the amount of source components is required. This is equivalent to the estimation of the rank of NMF factorization. If there are too few basis vectors, the algorithm may fail and miss important information. On the other hand, if there are too many basis vectors, an additional step is required to select the correct dimensionality from the available options. The best performance is achieved when the number of basis vectors and the number of notes are equal.

The rank estimation algorithm under the consideration include SVT-SURE which has limited flexibility in terms of soft-thresholding. The proposed algorithm aims to introduce SURE in the context of WSVT (WSVT-SURE) and construct its risk function to determine the proper threshold parameter. The minima of the risk function is going to determined via the exhaustive search and the weight vector for the proposed WSVT-SURE algorithm is going to be modeled specifically for the task of interest.

Also, this work introduces second algorithm for the proposed WSVT-SURE which is directed towards solving the issue of high computational complexity emerging from the exhaustive search approach. Namely, the gradient based optimization algorithm for the smooth approximation of WSVT-SURE (GWSVT-SURE) is derived. The smooth approximation is required because the risk function of the proposed WSVT-SURE contains multiple discontinuous functions which does not allow to derive its gradient. The approximation itself is conducted via replacing non-smooth functions with their smooth approximations such as smooth maximum unit and sigmoid function.

1.4 Report Organization

Chapter 2 provides a background study for fundamental NMF algorithms with their derivations as well the preliminary study needed for developing the proposed algorithms. Chapter 3 provides a detailed derivations for the proposed algorithms.

For the evaluation of the proposed algorithm, experimental setup, paradigm for parameter choice as well as simulation results for practical applications are provided in Chapter 4. Finally, In Chapter 5, concluding remarks and ideas for the future work are presented.

Chapter 2

Literature Review and Related Work

2.1 Multiplicative Update (MU) Nonnegative Matrix Factorization (NMF)

Multiplicative Update NMF (MU-NMF) is a widely used family of NMF algorithms for low rank matrix factorization. The term Multiplicative Update refers to the specific update rules used in the iterative optimization process which employs multiplicative rather than additive updates to iteratively optimize the factor matrices. The paramount importance of this family of algorithms lies in keeping the updated matrices to be nonnegative or designing MU algorithms to keep the prior assumptions true.

Referring to the equation (1.2), NMF problem is defined as an approximate decomposition of a given nonnegative data matrix representation $\boldsymbol{V} \in \mathbb{R}^{M \times N}$ into nonnegative matrices $\boldsymbol{W}, \boldsymbol{H}$ and computation of NMF is equivalent to the constrained optimization of the form:

$$\{\boldsymbol{W}, \boldsymbol{H}\} = \underset{\boldsymbol{W},\boldsymbol{H} \geq 0}{\operatorname{argmin}} \mathcal{D}(\boldsymbol{V}; \boldsymbol{W}, \boldsymbol{H}) \tag{2.1}$$

where $\mathcal{D}$ represents the cost function. The problem under consideration is to optimize alternatively $\boldsymbol{W}$ or $\boldsymbol{H}$ matrices while keeping other fixed. This is done due to the fact that alternative optimization imposes convexity as it transforms problem into Nonnegative Least Squares (NNLS) problem [37]. Hence, MU-NMF is defined as the pair-wise minimization:

$$\boldsymbol{W} = \underset{\boldsymbol{W} \geq 0}{\operatorname{argmin}} \mathcal{D}_1(\boldsymbol{V}; \boldsymbol{W}, \boldsymbol{H}), \quad \text{for fixed } \mathbf{H}, \tag{2.2}$$

$$\boldsymbol{H} = \underset{\boldsymbol{H} \geq 0}{\operatorname{argmin}} \mathcal{D}_2(\boldsymbol{V}; \boldsymbol{W}, \boldsymbol{H}). \quad \text{for fixed } \mathbf{W}. \tag{2.3}$$

The solution of MU-NMF is dependent on the choice of divergence metrics $\mathcal{D}(\boldsymbol{V}; \boldsymbol{W}, \boldsymbol{H})$

which defines the statistical distance between $\boldsymbol{WH}$ and $\boldsymbol{V}$. In other words, it shows how 'close' is factorization result to the original matrix.

One popular type of divergence metric is β-divergence [38] which encompasses multiple popular divergence metrics such as Frobenius norm (FN) ($\beta = 2$), Kullback-Leibler (KL) divergence ($\beta = 1$), Itakura-Saito (IS) divergence ($\beta = 0$). It is defined as:

$$\mathcal{D}_\beta(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}); & \beta \in \mathbb{R} \\ x\log(\frac{x}{y}) - x + y; & \beta = 1 \\ \frac{x}{y} - \log(\frac{x}{y}) - 1; & \beta = 0 \end{cases}, \tag{2.4}$$

where the x, y represent the optimization parameters. The solution to NMF problem via MU-NMF is then boils down to applying β divergence as a cost function. The update equations are derived by finding the global minima of the cost function by applying first order optimality conditions. First, define KL divergence as $\mathcal{D}_{\beta=1}(x|y) = \mathcal{D}_{KL}(x|y)$:

$$\mathcal{D}_{KL}(x|y) = x\log(\frac{x}{y}) - x + y. \tag{2.5}$$

Transforming the optimization parameters of equation (2.5) into the NMF variables $\boldsymbol{V}, \boldsymbol{WH}$ for x and y, respectively, the KL divergence takes the following form:

$$\mathcal{D}_{KL}(\boldsymbol{V};\boldsymbol{WH}) = \sum_m \sum_n v_{mn}\log(\frac{v_{mn}}{\boldsymbol{WH}|_{mn}}) - v_{mn} + \boldsymbol{WH}|_{mn}. \tag{2.6}$$

To satisfy the first order optimality conditions, the derivation of the gradient of KL-divergence (2.6) is required. The subsequent development of MU-NMF update equation is going to be performed for $\boldsymbol{W}$ factorization matrix.

$$\nabla_{\boldsymbol{W}}\mathcal{D}_{KL}(\boldsymbol{V};\boldsymbol{WH}) = \sum_m \sum_n x_{mn}\frac{\partial}{\partial w_{ij}}(\log(x_{mn}) - \log(\boldsymbol{WH}|_{mn})) + \sum_m \sum_n \frac{\partial}{\partial w_{ij}}\boldsymbol{WH}|_{mn}. \tag{2.7}$$

The resulting gradient of KL-divergence wrt. $\boldsymbol{W}$ is expressed as:

$$\nabla_{\boldsymbol{W}}\mathcal{D}_{KL}(\boldsymbol{V};\boldsymbol{WH}) = -\frac{\boldsymbol{V}}{\boldsymbol{WH}}\boldsymbol{H}^T + \boldsymbol{H}^T. \tag{2.8}$$

In order to obtain the update equation for $\boldsymbol{W}$ factorization matrix, corresponding gradient of the KL-divergence must be equated to zero.

$$-\frac{\boldsymbol{V}}{\boldsymbol{WH}}\boldsymbol{H}^T + \boldsymbol{H}^T = 0. \tag{2.9}$$

The final step is to express the equation in terms of $\boldsymbol{W}$, the resulting update rules with respect to $\boldsymbol{W}$ using the first order optimality conditions:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \otimes \frac{\frac{\boldsymbol{V}}{\boldsymbol{WH}}\boldsymbol{H}^T}{\boldsymbol{H}^T}. \tag{2.10}$$

Similar procedure is applied for the matrix $\boldsymbol{H}$, for which the update equation is expressed as:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \otimes \frac{\boldsymbol{W}^T\frac{\boldsymbol{V}}{\boldsymbol{WH}}}{\boldsymbol{W}^T}. \tag{2.11}$$

Combining the equations (2.10) and (2.11), the MU-NMF update rules for KL-divergence is derived as:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \otimes \frac{\boldsymbol{W}^T\frac{\boldsymbol{V}}{\boldsymbol{WH}}}{\boldsymbol{W}^T} \quad \text{for fixed } \mathbf{W}, \tag{2.12}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \otimes \frac{\frac{\boldsymbol{V}}{\boldsymbol{WH}}\boldsymbol{H}^T}{\boldsymbol{H}^T} \quad \text{for fixed } \mathbf{H}. \tag{2.13}$$

where $\otimes$ is defined as point-wise multiplication (Hadamard product) as well as division is also performed in the similar point-wise fashion. General MU-NMF for any β value is defined as:

$$\boldsymbol{H} \leftarrow \frac{\boldsymbol{H} \otimes \boldsymbol{W}^T(\boldsymbol{WH})^{\beta-2} \otimes \boldsymbol{V}}{\boldsymbol{W}^T(\boldsymbol{WH})^{\beta-2}} \quad \text{for fixed } \mathbf{W}, \tag{2.14}$$

$$\boldsymbol{W} \leftarrow \frac{((\boldsymbol{WH})^{\beta-2} \otimes \boldsymbol{V})\boldsymbol{V}}{(\boldsymbol{WH})^{\beta-1}\boldsymbol{H}^T} \quad \text{for fixed } \mathbf{H}. \tag{2.15}$$

The MU-NMF with general β divergence allows for flexibility in the choice of divergence measure and it typically performs well on different data types such as music, text, images etc. Also, it should be noted that before performing optimization (2.12-2.13), the variables $\boldsymbol{W}$ and $\boldsymbol{H}$ must be first initialized. Typically, random initialization is an appropriate choice for most of the applications.

2.2 Nonnegative Least Squares NMF

2.2.1 Alternating Least Squares NMF algorithm

Alternating Least Squares (ALS) NMF is a fast and efficient algorithm that mitigates the problem of factorizing a large scale data [39]. The main difference between ALS-NMF and MU-NMF is that latter solves the set of nonnegative least squares problems. In other words, at each iteration algorithm solves two constrained optimization problems for each of the matrices $\boldsymbol{W}, \boldsymbol{H}$.

Following the definition (1.2), ALS-NMF problem is defined as the set of minimization problems, where the Euclidean distance cost function is defined as:

$$\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W},\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_F^2, \quad \boldsymbol{W},\boldsymbol{H} \geq 0 \tag{2.16}$$

where $\|\cdot\|_F$ represents the Frobenius norm. Similar to the (2.2-2.3), at each iteration, ALS-NMF solves pair of optimization problems, in the alternating manner. The optimization is performed on the basis of Euclidean distance cost functions, expressed as:

$$\boldsymbol{W}^{(k+1)} = \arg\min_{W\geq 0} \|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}^{(k)}\|_F^2 \tag{2.17}$$

$$\boldsymbol{H}^{(k+1)} = \arg\min_{H\geq 0} \|\boldsymbol{V}^T-\boldsymbol{H}^T[\boldsymbol{W}^{(k+1)}]^T\|_F^2. \tag{2.18}$$

Instead of using gradient descent method, the solution could be estimated directly by Karush-Kuhn-Tucker (KKT) optimality conditions. This method returns the optimal solution $(\boldsymbol{W^*},\boldsymbol{H^*})$ for ALS-NMF optimization problem. KKT conditions are defined as:

1. Dual feasibility: $\boldsymbol{W^*} \geq 0$, $\boldsymbol{H^*} \geq 0$.
2. Primal Feasibility: $\nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W^*H^*}) \geq 0$, $\nabla_{\boldsymbol{H}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W^*H^*}) \geq 0$.
3. Comp. Slackness: $\boldsymbol{W^*} \otimes \nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W^*H^*}) = 0$, $\boldsymbol{H^*} \otimes \nabla_{\boldsymbol{H}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W^*H^*}) = 0$

As an example, consider the optimization problem for the $\nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W}\boldsymbol{H})$. Note that squared Frobenius norm is expressed as the trace-product of matrices:

$$\begin{aligned}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W}\boldsymbol{H}) &= \frac{1}{2}\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_F^2 \\ &= \frac{1}{2}tr((\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H})^T(\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H})),\end{aligned} \tag{2.19}$$

where $tr()$ represent the trace of the matrix. The gradient for the update equations of $\boldsymbol{W}$ matrix is expressed as:

$$\begin{aligned}\nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W}\boldsymbol{H}) &= \frac{\partial}{\partial \boldsymbol{W}}\frac{1}{2}\{tr((\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H})^T(\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}))\} \\ &= \frac{\partial}{\partial \boldsymbol{W}}\frac{1}{2}\{tr(-\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{V}+\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{H}+\boldsymbol{V}^T\boldsymbol{V}-\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{H})\} \\ &= tr(-\boldsymbol{H}^T\boldsymbol{V}+\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T).\end{aligned} \tag{2.20}$$

Assuming that the optimal solution is not zero, the stationary points are determined by setting the component of the gradient equal to zero:

$$\nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W},\boldsymbol{H}) = -\boldsymbol{V}\boldsymbol{H}^T+\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T = 0. \tag{2.21}$$

The update rule for ALS-NMF with respect to factorization matrix is expressed as $\boldsymbol{W}$:

$$\boldsymbol{W} \leftarrow \boldsymbol{V}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}. \tag{2.22}$$

The resulting update equation corresponds to matrix $\boldsymbol{W}$, the similar approach is applied towards derivation of the update rule for $\nabla_{\boldsymbol{H}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W}\boldsymbol{H})$, the gradient component for matrix $\boldsymbol{H}$ is expressed as:

$$\nabla_{\boldsymbol{W}}\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W},\boldsymbol{H}) = -\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}, \tag{2.23}$$

by equating the gradient component to zero:

$$-\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} = 0, \tag{2.24}$$

the resulting update equation with respect to matrix $\boldsymbol{H}$ is expressed as:

$$\boldsymbol{H} \leftarrow (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{V}. \tag{2.25}$$

Combining the equations (2.22) and (2.25), the pair of update equations for ALS-NMF takes the following form:

$$\boldsymbol{W} \leftarrow \boldsymbol{V}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1} \tag{2.26}$$

$$\boldsymbol{H} \leftarrow (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{V}. \tag{2.27}$$

One could note that algorithm requires taking the inverse of matrices at each iteration. This could be mitigated by imposing the Moore-Penrose pseudo inverse in the update rules [40]. Another issue with ALS-NMF algorithm is that it often fails to converge to the global minima. Therefore, it is necessary to apply regularization terms or develop different optimization strategies.

2.2.2 Hierarchical Alternating Least Squares NMF

Hierarchical ALS (HALS) algorithm for NMF is a local optimization method that is based on updating columns of the input data matrix. Let the matrices $\boldsymbol{W}, \boldsymbol{H}$ are fixed except the j^{th} column, the cost function is then defined as:

$$\mathcal{D}_F(\boldsymbol{V};\boldsymbol{W}\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{V}^j - \sum_{j=1}^{L}\boldsymbol{w}_j\boldsymbol{h}_j^T\|_F^2, \tag{2.28}$$

where $\boldsymbol{V}^{(j)}$ represents the residue matrix:

$$\begin{aligned}\boldsymbol{V}^{(j)} &= \boldsymbol{V} - \sum_{k\neq j}^{L}\boldsymbol{w}_k\boldsymbol{h}_k^T \\ &= \boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}^T + \boldsymbol{w}_j\boldsymbol{h}_j^T.\end{aligned} \tag{2.29}$$

The optimization scheme is now carried out by alternating minimization of a set of cost functions. In other words, we perform optimization of factorization matrices columnwise for $j = 1, 2 \cdots, L$.

The derivation of an update equations are performed similar to ALS-NMF with KKT optimality conditions (2.16-2.18). Consider the gradient computation of $\nabla_{w_{(j)}} \mathcal{D}_F^{(j)}(\boldsymbol{V}^{(j)}; \boldsymbol{w}_j, \boldsymbol{h}_j^T)$ being derived as:

$$\begin{aligned} \nabla_{\boldsymbol{w}_{(j)}} \mathcal{D}_F^{(j)}(\boldsymbol{V}^{(j)}; \boldsymbol{w}_j, \boldsymbol{h}_j^T) &= \frac{\partial}{\partial \boldsymbol{w}_{(j)}} \{\frac{1}{2} tr((\boldsymbol{V} - \boldsymbol{w}_j \boldsymbol{h}_j^T)^T (\boldsymbol{V} - \boldsymbol{w}_j \boldsymbol{h}_j^T))\} \\ &= \frac{\partial}{\partial \boldsymbol{w}_{(j)}} \{\frac{1}{2} tr(-\boldsymbol{w}_j^T \boldsymbol{h}_j \boldsymbol{V} + \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{w}_j^T \boldsymbol{h}_j \boldsymbol{w}_j \boldsymbol{h}_j^T - \boldsymbol{V}^T \boldsymbol{w}_j \boldsymbol{h}_j^T)\} \\ &= \boldsymbol{w}_j \boldsymbol{h}_j^T \boldsymbol{h}_j - \boldsymbol{V}\boldsymbol{h}_j. \end{aligned} \tag{2.30}$$

The resulting gradient is now must be equated to zero:

$$\boldsymbol{w}_j \boldsymbol{h}_j^T \boldsymbol{h}_j - \boldsymbol{V}^{(j)} \boldsymbol{h}_j = 0. \tag{2.31}$$

Assuming that entries of $\boldsymbol{w}_j$ are positive $\forall j$ and nonzero, the update equations have the form:

$$\boldsymbol{w}_j \leftarrow \frac{1}{\boldsymbol{h}_j^T \boldsymbol{h}_j} [\boldsymbol{V}^{(j)^T} \boldsymbol{h}_j]. \tag{2.32}$$

Similar procedure is conducted for $\nabla_{\boldsymbol{h}_{(j)}} \mathcal{D}_F^{(j)}(\boldsymbol{V}^{(j)}; \boldsymbol{w}_j, \boldsymbol{h}_j^T)$, the gradient with respect to the matrix entries of $\boldsymbol{H}$ is expressed as:

$$\nabla_{\boldsymbol{h}_{(j)}} \mathcal{D}_F^{(j)}(\boldsymbol{V}^{(j)}; \boldsymbol{w}_j, \boldsymbol{h}_j^T) = \boldsymbol{w}_j^T \boldsymbol{w}_j \boldsymbol{h}_j - \boldsymbol{V}^{(j)^T} \boldsymbol{w}_j, \tag{2.33}$$

by assuming the nonnegativity and nonzero $\boldsymbol{h}_j$, equating the gradient to zero:

$$\boldsymbol{w}_j^T \boldsymbol{w}_j \boldsymbol{h}_j - \boldsymbol{V}^{(j)^T} \boldsymbol{w}_j = 0, \tag{2.34}$$

the resulting closed form expression for the update equation for $\boldsymbol{h}_j$ is expressed as:

$$\boldsymbol{h}_j \leftarrow \frac{1}{\boldsymbol{w}_j^T \boldsymbol{w}_j} [\boldsymbol{V}^{(j)^T} \boldsymbol{w}_j]. \tag{2.35}$$

Combining the equations (2.32) and (2.35), HALS NMF update equations takes the following form:

$$\begin{aligned} \boldsymbol{w}_j &\leftarrow \frac{1}{\boldsymbol{h}_j^T \boldsymbol{h}_j} [\boldsymbol{V}^{(j)^T} \boldsymbol{h}_j] \\ \boldsymbol{h}_j &\leftarrow \frac{1}{\boldsymbol{w}_j^T \boldsymbol{w}_j} [\boldsymbol{V}^{(j)^T} \boldsymbol{w}_j]. \end{aligned} \tag{2.36}$$

Generally, HALS-NMF algorithm is similar to the previously discussed ALS-NMF, however the "hierarchicity" of the algorithm comes from the set of iterative updates of the residue matrices. They (residuals) are linked to each other in the hierarchical manner, as seen in (2.26). HALS-NMF is suitable for large-scale factorization problems and it is known to have a better performance than ALS-NMF [41].

2.3 Projected Gradient Optimization for NMF

All of the presented algorithms in the previous sections are employ the multiplicative update rules which suffers from relatively slow convergence rates. One solution towards improving the previous algorithms is to update the factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ based on standart gradient descent adaptation scheme:

$$\boldsymbol{W}^{k+1} = \max(0, \boldsymbol{W}^k - \mu_k \nabla_{\boldsymbol{W}} \mathcal{D}(\boldsymbol{V}; \boldsymbol{W}^k, \boldsymbol{H}^k)), \tag{2.37}$$

$$\boldsymbol{H}^{k+1} = \max(0, \boldsymbol{H}^k - \mu_k \nabla_{\boldsymbol{H}} \mathcal{D}(\boldsymbol{V}; \boldsymbol{W}^k, \boldsymbol{H}^k)). \tag{2.38}$$

This algorithm is known as Projected Gradient NMF (PGD-NMF). As this update rule has an additive property, $\max(x, y)$ function is used to ensure the nonnegativity of the result. PGD-NMF algorithm is shown to have better converge rate than standard MU-NMF and it has less computational complexity [45].

PGD-NMF provides an approximate solution for NNLS problem (2.14-2.15). The gradient update rules for NNLS problem is expressed as:

$$\boldsymbol{W}^{(k+1)} = \max(0, \boldsymbol{W}^{(k)} - \mu_k(\boldsymbol{V}\boldsymbol{H}^{(k)^T} + \boldsymbol{W}^{(k)}\boldsymbol{H}^{(k)}\boldsymbol{H}^{(k)^T})), \tag{2.39}$$

$$\boldsymbol{H}^{(k+1)} = \max(0, \boldsymbol{H}^{(k)} - \mu_k(\boldsymbol{W}^{(k)^T}\boldsymbol{V} + \boldsymbol{W}^{(k)^T}\boldsymbol{W}^{(k)}\boldsymbol{H}^{(k)})), \tag{2.40}$$

where the gradient values of (2.28-2.29) are substituted with previously found gradient values for NNLS problem (2.21-2.22). The parameter μ_k represent the step-size parameter of a gradient descent. It has been deduced that for Landweber iterations [42], the feasible set for step-size parameter ensuring asymptotic convergence is $\mu \in (0, \mu_{\max})$, where $\mu_{\max}$ is expressed as [43]:

$$\mu_{\max} = \frac{2}{l_{\max}\{\boldsymbol{W}^T\boldsymbol{W}\}}, \tag{2.41}$$

where $l_{\max}\{\boldsymbol{W}^T\boldsymbol{W}\}$ represent the maximum eigenvalue of $\boldsymbol{W}^T\boldsymbol{W}$.

The summary of some of the fundamental algorithms for NMF is represented in Figure 2.1 in tabular form. Additionally, Convolutive NMF [44] algorithm is presented due to its range of applications in music, speech, text separation.

Algorithm	Cost function	Update Equations
NMF based on β divergence (MU-NMF)	$d_\beta = \begin{cases} \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}, \beta \in \mathbb{R}\{0,1\} \\ x\,log\left(\frac{x}{y}\right) - x + y\,, \beta = 1 \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1, \beta = 0 \end{cases}$	$\boldsymbol{H} \leftarrow \frac{\boldsymbol{H} \otimes \boldsymbol{W}^T[(\boldsymbol{WH})^{\beta-2} \otimes \boldsymbol{V}]}{\boldsymbol{W}^T(\boldsymbol{WH})^{\beta-2}}$ $\boldsymbol{W} \leftarrow \frac{[(\boldsymbol{WH})^{\beta-2} \otimes \boldsymbol{V}]\boldsymbol{H}^T}{(\boldsymbol{WH})^{\beta-1}\boldsymbol{H}^T}$
Alternating Least Squares NMF (ALS-NMF)	$D_F(\boldsymbol{V}\|\boldsymbol{WH}) = \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{WH}\|_F^2$	$\mathbf{W} = \left[\boldsymbol{V}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\right]_+$ $\mathbf{H} = \left[\left(\boldsymbol{W}\boldsymbol{W}^T\right)^{-1}\boldsymbol{W}^T\boldsymbol{V}\right]_+$
Hierarchical Alternating Least Squares NMF (HALS-NMF)	$D_F(\boldsymbol{V}\|\boldsymbol{WH}) = \frac{1}{2}\left\|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}^T\right\|_F^2 = \frac{1}{2}\left\|\boldsymbol{V} - \sum_{j=1}^{J} \boldsymbol{w}_j\boldsymbol{h}_j^T\right\|_F^2$	$\boldsymbol{h}_j = \frac{1}{\boldsymbol{w}_j^T\boldsymbol{w}_j}\left[(\boldsymbol{V}^j)^T\boldsymbol{w}_j\right]$ $\boldsymbol{w}_j = \frac{1}{\boldsymbol{h}_j^T\boldsymbol{h}_j}\left[(\boldsymbol{V}^j)^T\boldsymbol{h}_j\right]$
Projected Gradient NMF (PG-NMF)	$D_F(\boldsymbol{V}\|\boldsymbol{WH}) = \frac{1}{2}\|\boldsymbol{V} - \boldsymbol{WH}\|_F^2$	$\boldsymbol{H}^{k+1} = \boldsymbol{H}^k - \mu_H^k\left[\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{H} - \boldsymbol{W}^T\boldsymbol{V}\right]$ $\boldsymbol{W}^{k+1} = \boldsymbol{W}^k - \mu_W^k\left[\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{V}\boldsymbol{H}^T\right]$
Convolutive NMF (CNMF)	$d_\beta = \begin{cases} \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}, \beta \in \mathbb{R}\{0,1\} \\ x\,log\left(\frac{x}{y}\right) - x + y\,, \beta = 1 \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1, \beta = 0 \end{cases}$	$\boldsymbol{H} \leftarrow \boldsymbol{H} \otimes \frac{\boldsymbol{W}_t^T\left[\frac{\boldsymbol{V}}{\tilde{\boldsymbol{V}}}\right]^{\leftarrow t}}{\boldsymbol{W}_t^T}$ $\boldsymbol{W}_t \leftarrow \boldsymbol{W}_t \otimes \frac{\left[\frac{\boldsymbol{V}}{\tilde{\boldsymbol{V}}}\right](\boldsymbol{H}^T)^{\rightarrow t}}{(\boldsymbol{H}^T)^{\rightarrow t}}$

Figure 2.1: Summary of fundamental NMF algorithms

2.4 Singular Value Thresholding-Stein's Unbiased Risk Estimate (SVT-SURE)

As the primary research objective of this work is to estimate the number of musical notes in the recording or equivalently, the rank of NMF, the development of a precise estimator is required. For this task, the convex rank reduction technique represented by SVT is used in tandem with SURE. The role of latter is to control the amount of reduced rank, thus by achieving precise estimation. The reader must note that both mathematical tools are completely different at the fundamental level, and it is a complicated task to link the SVT with SURE in a closed form expression. In fact, up to the current literature, some estimators such as Robust PCA has not been developed for SURE [46].

2.4.1 Singular Value Thresholding (SVT)

Consider the denoising problem being represented in the form of NMF. Let the input data matrix $\boldsymbol{V} \in \mathbb{R}^{M \times N}$ is corrupted with noise matrix $\boldsymbol{E} \in \mathbb{R}^{M \times N}$ of noise variance τ_e^2 for which the NMF model is expressed as:

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{W}\boldsymbol{H} + \boldsymbol{E} \\ &= \boldsymbol{V_0} + \boldsymbol{E}, \end{aligned} \tag{2.42}$$

where $\boldsymbol{E}$ represent Gaussian distributed noise with variance τ_e^2:

$$\boldsymbol{E} \sim \mathcal{N}(0, \tau_e^2). \tag{2.43}$$

The objective is to recover the matrix $\boldsymbol{V_0}$ from noisy observation, achieving (ideally) $\boldsymbol{V_0} = \boldsymbol{V}$, assuming the nonnegativity and low rank property of matrix $\boldsymbol{V_0}$. The essence of SVT lies on the processing of the singular values of a matrix. For the further tractable evaluation of SVT, the definition of Singular Value Decomposition (SVD) must be stated. Assuming real and nonnegative matrix $\boldsymbol{V}$, the SVD is being defined as:

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{P}^T \\ &= \sum_{i=1}^{\min(M,N)} \sigma_i \boldsymbol{u}_i \boldsymbol{p}_i^T, \end{aligned} \tag{2.44}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix of singular values σ_i, $\boldsymbol{U}$ is a left orthogonal matrix with $\boldsymbol{u}_i$ being its i^{th} vector, and $\boldsymbol{P}$ is a right orthogonal matrix with $\boldsymbol{p}_i$ being its its i^{th} vector. The SVT could be defined by imposing soft-thresholding operator $\mathcal{S}(\cdot)$

on the singular value matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{V}$.

$$SVT(\boldsymbol{V}) = \boldsymbol{U}\mathcal{S}(\boldsymbol{\Sigma})\boldsymbol{P}^T \tag{2.45}$$
$$= \sum_{i=1}^{\min(M,N)} \max(\sigma_i - \lambda, 0)\boldsymbol{u}_i\boldsymbol{p}_i^T.$$

SVT model is regarded as convex alternative for rank minimization problem where the rank reduction is carried out by the maximum function and λ threshold parameter. The reader must note that rank of the matrix could be represented by the number of non-zero singular values. Therefore, shrinkage the singular values by the correct amount of λ will result in the precise estimation of the rank. The problem stated in (2.39) is solved by constrained optimization resulting in [47]:

$$\mathcal{D}_\lambda(\boldsymbol{V}) = \arg\min_{\boldsymbol{V_0}}\{\frac{1}{2}\|\boldsymbol{V} - \boldsymbol{V_0}\|_F^2 + \lambda\|\boldsymbol{V_0}\|_w\}, \tag{2.46}$$

where $\|\boldsymbol{V_0}\|_w$ denotes the nuclear norm being computed as the sum of singular values σ_i for $i = 1, \ldots, \min(M, N)$ of a matrix $\boldsymbol{V}$. In order to determine the $\boldsymbol{V}_0$, the iterative procedure for shrinkage is applied. Firstly, the threshold parameter λ and a sequence of step-size parameters μ_k are fixed. Then, for $k = 1, 2, 3, \ldots$ the following procedure is performed:

$$\boldsymbol{V}^{(k)} = \mathcal{D}_\lambda(\boldsymbol{V}_0^{(k-1)}), \tag{2.47}$$
$$\boldsymbol{V_0}^{(k)} = \boldsymbol{V_0}^{(k-1)} + \mu_k\mathcal{P}_\Phi(\boldsymbol{M} - \boldsymbol{V}^{(k)}) \tag{2.48}$$

This iterative scheme is specifically designed for matrix completion problem, $\boldsymbol{M}$ being the unknown matrix and $\mathcal{P}_\Phi$ is a orthogonal projector on the random subset Φ.

2.4.2 Stein's Unbiased Risk Estimate (SURE)

The key question stands on how to choose the λ parameter to maximize the approximation? The small value for λ will cause overfitting whereas the large values for this parameter will retain the noise. The trade-off could be achieved by finding optimal λ_0 parameter through minimizing mean-square error (MSE):

$$R_\lambda = \mathbb{E}\{\|\boldsymbol{V_0} - \hat{\boldsymbol{V}}_\mathbf{0}\|_F^2\}, \tag{2.49}$$

where the $\mathbb{E}\{\cdot\}$ denotes the expectation of quantity inside and $\hat{\boldsymbol{V}}_\mathbf{0}$ is the estimate of the noiseless observation matrix.. The fundamental problem with MSE minimization is that the true estimation $\boldsymbol{V_0}$ is unknown, so a calculable unbiased estimator of the risk is required. One of the approach towards resolving this is to derive a

calculable unbiased estimator of the risk depending purely on the observed data. First, rewrite the (8) in the vector form:

$$\boldsymbol{V} = \boldsymbol{W}\boldsymbol{h}(n) + \boldsymbol{e}(n) = \boldsymbol{v}_0(n) + e(n). \tag{2.50}$$

To derive an unbiased estimator, expand (2.43) as:

$$\begin{aligned} R_\lambda &= \mathbb{E}\{\|(\boldsymbol{V} - \boldsymbol{V_0}) - (\boldsymbol{V} - \hat{\boldsymbol{V}}_{\boldsymbol{0}})\|_F^2\} \\ &= J\sigma^2 + \mathbb{E}\{\|(\boldsymbol{v}(n) - \hat{\boldsymbol{v}}(n)\|^2\} \\ &\quad - 2\mathbb{E}\{(\boldsymbol{v}(n) - \hat{\boldsymbol{v}}_0(n))^T(\boldsymbol{v}(n) - \boldsymbol{v}_0(n))\}, \end{aligned} \tag{2.51}$$

where $(\boldsymbol{V} - \boldsymbol{V_0})$ and $(\boldsymbol{V} - \hat{\boldsymbol{V}}_{\boldsymbol{0}})$ represent the error and noise matrices, respectively. The last term in (2.51) could be defined as the degree of freedom, being expressed as the covariance between $\boldsymbol{v}$ and $\hat{\boldsymbol{v}_0}$:

$$\begin{aligned} \mathrm{df} &= \frac{1}{\sigma^2}\mathrm{Cov}(\boldsymbol{v}, \hat{\boldsymbol{v}_0}) \\ &= \mathbb{E}\{(\boldsymbol{v}(n) - \hat{\boldsymbol{v}}_0(n))^T(\boldsymbol{v}(n) - \boldsymbol{v}_0(n))\}, \end{aligned} \tag{2.52}$$

Assuming the absolute continuity of soft thresholding operator into account, it is valid and enough to satisfy the Stein's Lemma [48] in order to derive the statistical estimate of a MSE. The resultant equation is refereed as SURE risk function:

$$R_\lambda = \|\hat{\boldsymbol{V}}_{\boldsymbol{0}} - \boldsymbol{V}\|_F^2\} + 2\tau_e^2\mathrm{df}(\hat{\boldsymbol{V}}_{\boldsymbol{0}}) - J\tau_e^2. \tag{2.53}$$

The risk function of this form contains the estimation matrix $\hat{\boldsymbol{V}}_{\boldsymbol{0}}$ which is this study referred to SVT estimate $\mathcal{D}_\lambda(\boldsymbol{V})$. Therefore, the risk function for the further analysis must take the form of:

$$R_\lambda = \|\{\mathcal{D}_\lambda(\boldsymbol{V}) - \boldsymbol{V}\|_F^2\} + 2\tau_e^2\mathrm{df}(\mathcal{D}_\lambda(\boldsymbol{V})) - J\tau_e^2, \tag{2.54}$$

where 'df' represents the divergence term of an unbiased estimator and J is a real positive scalar. Derivation of the risk function now concerns finding the expression for noise and divergence term.

The noise term is found by taking difference between SVT function (2.39) and SVD of a matrix $\boldsymbol{V}$ (2.38), which results in the piece-wise continuous function written as maximum function:

$$\|\mathcal{D}_\lambda(\boldsymbol{V}) - \boldsymbol{V}\|_F^2 = \sum_{i=1}^{\min(M,N)} \min(\sigma_i^2, \lambda^2). \tag{2.55}$$

Furthermore, the divergence term of a risk function has been derived for both real and complex cases as well as for general spectral functions in [21]. Assuming the

real data, the divergence term of a risk function can be expressed as:

$$\begin{aligned}\mathrm{df}(\mathcal{D}_\lambda(\boldsymbol{V})) = |M-N| \sum_{i=1}^{\min(M,N)} \max(1-\frac{\lambda}{\sigma_i},0) \\ + \sum_{i=1}^{\min(M,N)} \mathbb{I}(\sigma_i > \lambda) \\ + 2\sum_{i\neq j,i,j=1}^{\min(M,N)} \frac{\sigma_i \max(\sigma_i-\lambda,0)}{\sigma_i^2-\sigma_j^2},\end{aligned} \tag{2.56}$$

where $\mathbb{I}(\cdot)$ represents indicator function. Combining the (2.49) and (2.48), we will deduce the complete closed form expression of risk function for SVT. Subsequent rank estimation is carried out by iterating through λ parameters and deducing the minimum of a risk function.

$$\lambda_0 = \underset{\lambda}{\operatorname{argmin}}\, R_\lambda \tag{2.57}$$

There are two main problems with SVT-SURE that must be addressed. First of all, there is no possibility for one to control how many singular values are to be compressed. The shrinkage is applied equally which makes SVT-SURE algorithm inflexible rank estimator when considering different applications. Also, the optimal threshold parameter λ_0 is found by iterating over multiple candidate ranks, which increases the complexity of SVT-SURE. Moreover, in some applications rank estimation might be highly dependent on λ_0, which means that iterations must be conducted on almost continuous space of λ.

Chapter 3

Proposed Algorithms

3.1 Proposed Weighted SVT-SURE (WSVT-SURE) Algorithm

The main motivation behind introducing the weighted variant of SVT-SURE lies in the fact that it produces the uniform shrinkage of singular values. In other words, there is a need to introduce the flexibility for an estimator. One way of possessing the control over the estimator has been presented in [36], this work investigated a weighted variant of a SVT (WSVT) in which the threshold parameter is scaled with a weight vector producing non-uniform shrinkage. However, the risk function must be introduced for weighted thresholding estimator in order to achieve the desired rank estimation results. The proposed method will exploit the same derivation steps as SVT-SURE and its theoretical foundation will rely on [34].

Using the definition of recovery problem given in (2.37) and utilizing the similar assumptions, $\boldsymbol{V_0}$ matrix recovery is carried out by solving:

$$\mathcal{D}_{w,\lambda}(\boldsymbol{V}) = \arg\min_{\boldsymbol{V_0}}\{\frac{1}{2}\|\boldsymbol{V}-\boldsymbol{V_0}\|_F^2 + \lambda\|\boldsymbol{V_0}\|_{w,*}\}, \tag{3.1}$$

where $\|\boldsymbol{V_0}\|_{w,*}$ denotes the weighted nuclear norm being computed as the weighted sum of singular values $w_i\sigma_i$ for $i = 1,\ldots,\min(M,N)$ of a matrix $\boldsymbol{V}$. The solution of (3.2) could be interpreted as λ-weighted soft thresholding operation on singular values:

$$\mathcal{D}_{w,\lambda}(\boldsymbol{V}) = \sum_{i=1}^{\min(M,N)} \max(\sigma_i - w_i\lambda, 0)\boldsymbol{u}_i\boldsymbol{p}_i^T. \tag{3.2}$$

Similarly, the rank estimation is carried out by applying weighted soft thresholding to the singular values with the number of non-zero singular values represent the rank of the matrix. The selection of the optimal λ parameter is carried out by minimizing the unbiased estimator of the form:

$$R_{w,\lambda} = \|\mathcal{D}_{w,\lambda}(\boldsymbol{V}) - \boldsymbol{V}\|_F^2 + 2\tau_e^2\mathrm{df}(\mathcal{D}_{w,\lambda}(\boldsymbol{V})) - J\tau_e^2 \tag{3.3}$$

where the error term is $\|\mathcal{D}_{w,\lambda}(\boldsymbol{V}) - \boldsymbol{V}\|_F^2$ is expressed as:

$$\mathcal{D}_{w,\lambda}(\boldsymbol{V}) - \boldsymbol{V} = \sum_{i=1}^{\min(M,N)} \max(\sigma_i - w_i\lambda, 0)\boldsymbol{u}_i\boldsymbol{p}_i^T - \sigma_i\boldsymbol{u}_i\boldsymbol{p}_i^T \tag{3.4}$$

which is the difference between SVT estimate and SVD representation of the data matrix. In order to derive the closed form expression for the error term, two different cases must be considered:

$$\sum_{i=1}^{\min(M,N)} \max(\sigma_i - w_i\lambda, 0)\boldsymbol{u}_i\boldsymbol{p}_i^T - \sigma_i\boldsymbol{u}_i\boldsymbol{p}_i^T = \begin{cases} \sum_{i=1}^{\min(M,N)} w_i\lambda_i\boldsymbol{u}_i\boldsymbol{p}_i^T; & \sigma_i \geq w_i\lambda \\ -\sum_{i=1}^{\min(M,N)} \sigma_i\boldsymbol{u}_i\boldsymbol{p}_i^T; & \sigma_i < w_i\lambda \end{cases} \tag{3.5}$$

Using the fact that Frobenius norm of SVD is the squared sum of singular values, we could combine (3.6) into a compact representation which is expressed as:

$$\|\mathcal{D}_{w,\lambda}(\boldsymbol{V}) - \boldsymbol{V}\|_F^2 = \sum_{i=1}^{\min(M,N)} \min(\sigma_i^2, w_i^2\lambda^2). \tag{3.6}$$

Substitution of (3.6) into the risk function (3.3) results in:

$$R_{w,\lambda} = \sum_{i=1}^{\min(M,N)} \min(\sigma_i^2, w_i^2\lambda^2) + 2\tau_e^2\mathrm{df}(\mathcal{D}_{w,\lambda}(\boldsymbol{V})) - J\tau_e^2. \tag{3.7}$$

The algorithm derivation boils down towards generalization of the divergence term of SVT [34] to WSVT. First, define the spectral function of WSVT and its derivative similar to (3.3), as:

$$f_i(\sigma_i) = \max(\sigma_i - w_i\lambda, 0), \tag{3.8}$$

$$f_i'(\sigma_i) = \begin{cases} 1; & (\sigma_i - w_i\lambda) \geq 0, \\ 0; & \text{otherwise} \end{cases} \tag{3.9}$$

where $i = 1, 2, \cdots, \min(M, N)$. In order to introduce the SURE for WSVT, we must ensure that the theoretical analysis provided in [34] for SVT can be generalized for WSVT. First of all, we make an assumption that WSVT is Lipschitz continuous, although it is not generally convex, the choice of weights in non-ascending order transforms the non-convex WSVT to a convex problem [36].

Theorem 1: *Let $\mathcal{F}$ be a space of simple (no repeating singular values), full rank matrices with $\sigma_i \neq w_i\lambda$ for $i = 1, 2, \cdots, \min(M, N)$, the mapping of WSVT is differentiable over $\mathcal{F}$.*

Proof: The spectral function (3.9) is differentiable (resulting in (3.10)) at any matrix point of space $\mathcal{F}$. Simple and full rank matrix assumptions ensure that WSVT is differentiable by [IV.1,24].

The differentiability property of WSVT allows us to derive the closed form expression for differential of spectral function [24]. Furthermore, it enables to employ the [IV.3,24] in which the spectral functions $f_i(\sigma_i), f_i'(\sigma_i)$ in the divergence term are replaced with (3.9) and (3.10). The resulting expression for the divergence term of WSVT-SURE becomes:

$$\begin{aligned}\mathrm{df}(\mathcal{D}_{w,\lambda}(\boldsymbol{Y})) = 2 &\sum_{i\neq j, i,j=1}^{\min(M,N)} \frac{\sigma_i \max(\sigma_i - w_i\lambda, 0)}{\sigma_i^2 - \sigma_j^2} \\ &+ \sum_{i=1}^{\min(M,N)} |M-N| \max(1 - \frac{w_i\lambda}{\sigma_i}, 0) \\ &+ \mathbb{I}(\sigma_i > w_i\lambda),\end{aligned} \tag{3.10}$$

which in combination with (3.8) gives the complete risk function for the proposed algorithm. The remaining problem concerns choosing the correct weight vector which is conventionally expressed as an inverse of the singular values to enhance the large singular values and remove the effect of smaller ones:

$$w_i = \frac{c}{\sigma_i + \delta} + s, \tag{3.11}$$

where c and s are consant parameters (their selection will be explained later) and δ is a small positive constant to avoid division by zero.

A flow chart of the proposed WSVT-SURE algorithm is given in Fig. 3.1. Algorithm first transforms the time-domain signal to the magnitude spectrogram representation via Short Time Fourier Transform (STFT) after which it computes its SVD. Then, we manually initialize the weight parameter c_0 and compute the weight vector $\boldsymbol{w}$ with subsequent multiplication with the set of λ thresholds. The thresholds are then fed into the risk function iteratively and the optimal threshold is chosen such that it attains minima of the risk function. Furthermore, the rank of the matrix is then estimated with soft-thresholding operator until the estimated rank does not reach the desirable result. Otherwise, weight vector parameter is adjusted and fed back to the algorithm.

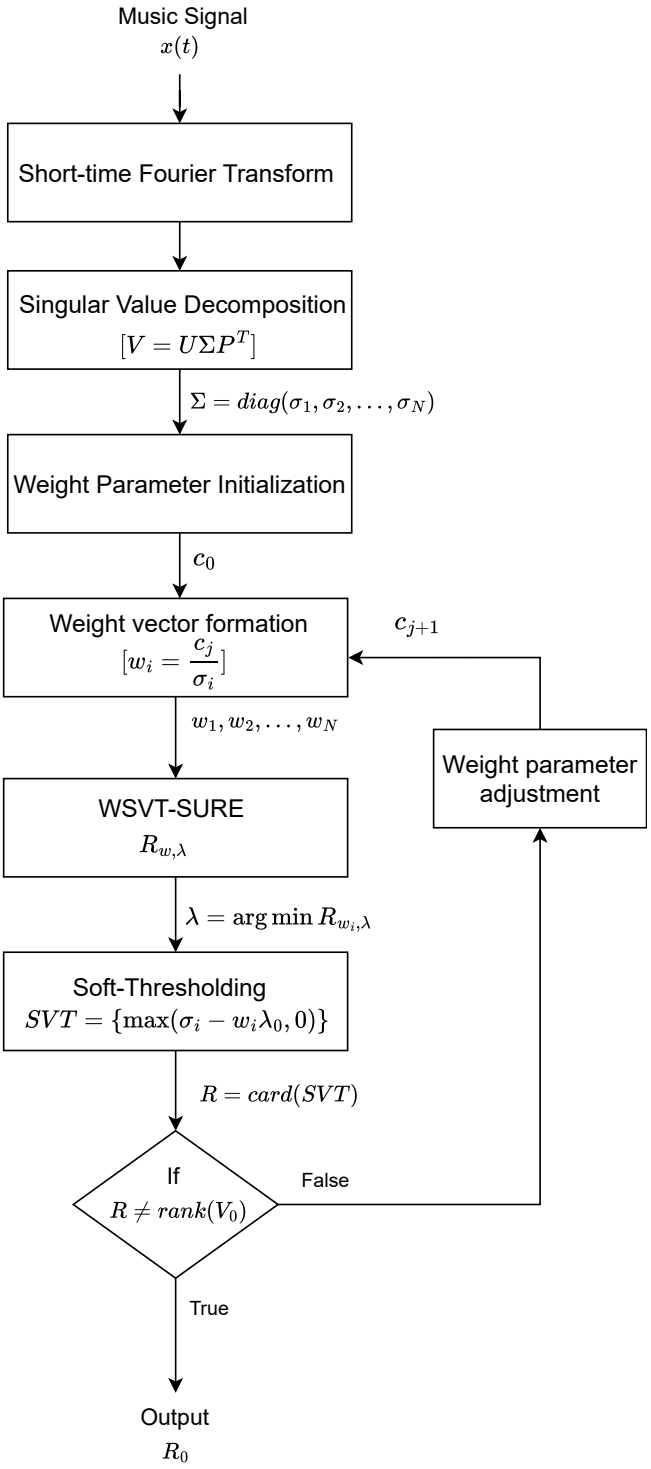

Figure 3.1: The flow chart for proposed WSVT-SURE algorithm.

3.2 Proposed Gradient WSVT-SURE Algorithm

3.2.1 Attempts for Gradient Optimization of the Risk Function

To mitigate the high computational complexity from the exhaustive search of WSVT-SURE, it is proposed to implement the gradient descent optimization of the risk function (3.8):

$$\nabla R_{w,\lambda} = \sum_{i=1}^{\min(M,N)} \frac{\partial}{\partial \lambda}\{\min(\sigma_i^2, w_i^2\lambda^2)\} + 2\tau_e^2 \frac{\partial}{\partial \lambda}\{\mathrm{df}(\mathcal{D}_{w,\lambda}(\boldsymbol{V}))\}. \tag{3.12}$$

The resulting expression for the gradient of the risk function takes the form:

$$\nabla R_{w,\lambda} = \sum_{i=1}^{\min(M,N)} 2\lambda w_i^2 + 2|M-N| \sum_{i=1}^{\min(M,N)} -\frac{w_i}{\sigma_i} + 4 \sum_{i,j=1, i\neq j}^{\min(M,N)} -\frac{w_i\sigma_i}{(\sigma_i^2 - \sigma_j^2)}. \tag{3.13}$$

However, there is a issue of non-differentiability of the risk function because it contains non-smooth parts such as maximum and indicator functions in (3.5-3.6). Although, it must be noted that minimum, maximum and indicator functions are weakly differentiable, it does not satisfy the requirements for gradient optimization. In other words, the risk function gradient (3.14) is mathematically plausible, it does not track the global minima and does not fit for optimization problems.

3.2.2 GWSVT-SURE via Smooth Approximations

As noted in [34], the risk function as described in (3.10) is not differentiable in a strict sense which restricts the use of gradient method directly with the unbiased estimator. However, one way of mitigating this issue is to utilize the smooth maximum unit (SMU) [49] and sigmoid function to convert the risk function from non differentiable to smooth form, which thus allows to apply gradient descent optimization. Maximum and indicator functions are approximated, respectively as:

$$\max(x,y) \approx \frac{x+y+\sqrt{(x-y)^2+\epsilon}}{2}, \tag{3.14}$$

$$\mathbb{I}(x>y) \approx \frac{1}{1+e^{-K(x-y)}}, \tag{3.15}$$

where ϵ and K represent the parameters controlling the accuracy of approximation such that for $\epsilon \to 0$, $K \to \infty$ the equality takes place. Transformations of (3.14) and (3.15) allows to find the derivative of the risk function with respect to threshold

parameter λ. The transformed risk function takes the form of:

$$\begin{aligned}\hat{R}_{w,\lambda} = &- \sum_{i=1}^{\min(M,N)} \frac{-\sigma_i^2 - w_i^2\lambda^2 + \sqrt{(-\sigma_i^2 + w_i^2\lambda^2)^2 + \epsilon}}{2} \\ &+ 2\tau_e^2|M-N| \sum_{i=1}^{\min(M,N)} \frac{1 - \frac{w_i^2\lambda}{\sigma_i} + \sqrt{(1 - \frac{w_i^2\lambda}{\sigma_i})^2 + \epsilon}}{2} \\ &+ 2 \sum_{i\neq j,i,j=1}^{\min(M,N)} \frac{\sigma_i}{\sigma_i^2 - \sigma_j^2} \left\{ \frac{\sigma_i - w_i\lambda + \sqrt{(-\sigma_i + w_i\lambda)^2 + \epsilon}}{2} \right\} \\ &+ \sum_{i=1}^{\min(M,N)} \frac{1}{1 + e^{-K(\sigma_i - w_i\lambda)}}\end{aligned} \tag{3.16}$$

Risk function defined in (3.16) is strictly differentiable and satisfies the requirements for the gradient optimization given that appropriate parameters for K, ϵ are chosen. The derivative of (3.16) is going to be computed with respect to the λ parameter:

1.

$$\begin{aligned}&- \frac{\partial}{\partial\lambda} \sum_{i=1}^{\min(M,N)} \frac{-\sigma_i^2 - w_i^2\lambda^2 + \sqrt{(-\sigma_i^2 + w_i^2\lambda^2)^2 + \epsilon}}{2} = \\ &= \sum_{i=1}^{\min(M,N)} \frac{\lambda w_i^2(\lambda^2 w_i^2 - {\sigma_i}^2 - \sqrt{(\lambda^2 w_i^2 - {\sigma_i}^2)^2 + \epsilon}}{\sqrt{(\lambda^2 w_i^2 - {\sigma_i}^2)^2 + \epsilon}}\end{aligned} \tag{3.17}$$

2.

$$\frac{\partial}{\partial\lambda} \sum_{i=1}^{\min(M,N)} \frac{1 - \frac{w_i^2\lambda}{\sigma_i} + \sqrt{(1 - \frac{w_i^2\lambda}{\sigma_i})^2 + \epsilon}}{2} = \sum_{i=1}^{\min(M,N)} \frac{w_i(\lambda w_i - \sigma_i)}{{\sigma_i}^2\sqrt{\frac{(\sigma_i - \lambda w_i)^2}{{\sigma_i}^2} + \epsilon}} - \frac{w_i}{\sigma_i}$$

3.

$$\frac{\partial}{\partial\lambda} \sum_{i\neq j,i,j=1}^{\min(M,N)} \frac{\sigma_i - w_i\lambda + \sqrt{(-\sigma_i + w_i\lambda)^2 + \epsilon}}{2} = \sum_{i\neq j,i,j=1}^{\min(M,N)} -w_i - \frac{w_i(\sigma_i - \lambda w_i)}{\sqrt{(\sigma_i - \lambda w_i)^2 + \epsilon}}$$

4.

$$\frac{\partial}{\partial\lambda} \sum_{i,j=1}^{\min(M,N)} \frac{1}{1 + e^{-K(\sigma_i - w_i\lambda)}} = \sum_{i=1}^{\min(M,N)} \frac{Kw_i e^{-K(-\lambda w_i + \sigma_i)}}{(1 + e^{-K(-\lambda w_i + \sigma_i)})^2}$$

Combining the derivatives of (3.18, 1-4), resulting gradient of a smooth approximate of a risk function is expressed as :

$$\begin{aligned}\nabla \hat{R}_{w,\lambda} \approx \tau_e |M-N| & \sum_{i=1}^{\min(M,N)} \left(\frac{w_i(\lambda w_i - \sigma_i)}{{\sigma_i}^2 \sqrt{\frac{(\sigma_i - \lambda w_i)^2}{{\sigma_i}^2} + \epsilon}} - \frac{w_i}{\sigma_i} \right) \\ & - \sum_{i=1}^{\min(M,N)} \frac{\lambda w_i^2 (\lambda^2 w_i^2 - {\sigma_i}^2 - \sqrt{(\lambda^2 w_i^2 - {\sigma_i}^2)^2 + \epsilon}}{\sqrt{(\lambda^2 w_i^2 - {\sigma_i}^2)^2 + \epsilon}} \\ & + 2\tau_e \sum_{i \neq j, i, j=1}^{\min(M,N)} \left(-w_i - \frac{w_i(\sigma_i - \lambda w_i)}{\sqrt{(\sigma_i - \lambda w_i)^2 + \epsilon}} \right) \frac{\sigma_i}{\sigma_i^2 - \sigma_j^2} \\ & - 2\tau_e \sum_{i=1}^{\min(M,N)} \frac{K w_i e^{-K(-\lambda w_i + \sigma_i)}}{(1 + e^{-K(-\lambda w_i + \sigma_i)})^2}\end{aligned}$$

Threshold optimization is carried out by standart gradient descent scheme:

$$\lambda^{(n+1)} = \lambda^{(n)} - \mu \nabla R_{w,\lambda}, \tag{3.18}$$

where $\lambda^{(n+1)}, \lambda^{(n)}$ represents the next and current iterates of threshold parameter, μ is defined as step-size controlling the learning rate of an algorithm.

Fig. 3.2. represents the flow chart of GWSVT-SURE algorithm. Similarly to WSVT-SURE, algorithm generates the magnitude spectrogram and SVD of data matrix after which initializes gradient optimization, and accuracy control parameters. Then, after manual initialization of weight vector, we compute the gradient of the risk function and perform the optimization step. Finally, if the stopping criteria is met, algorithm performs soft thresholding and outputs the rank of the matrix.

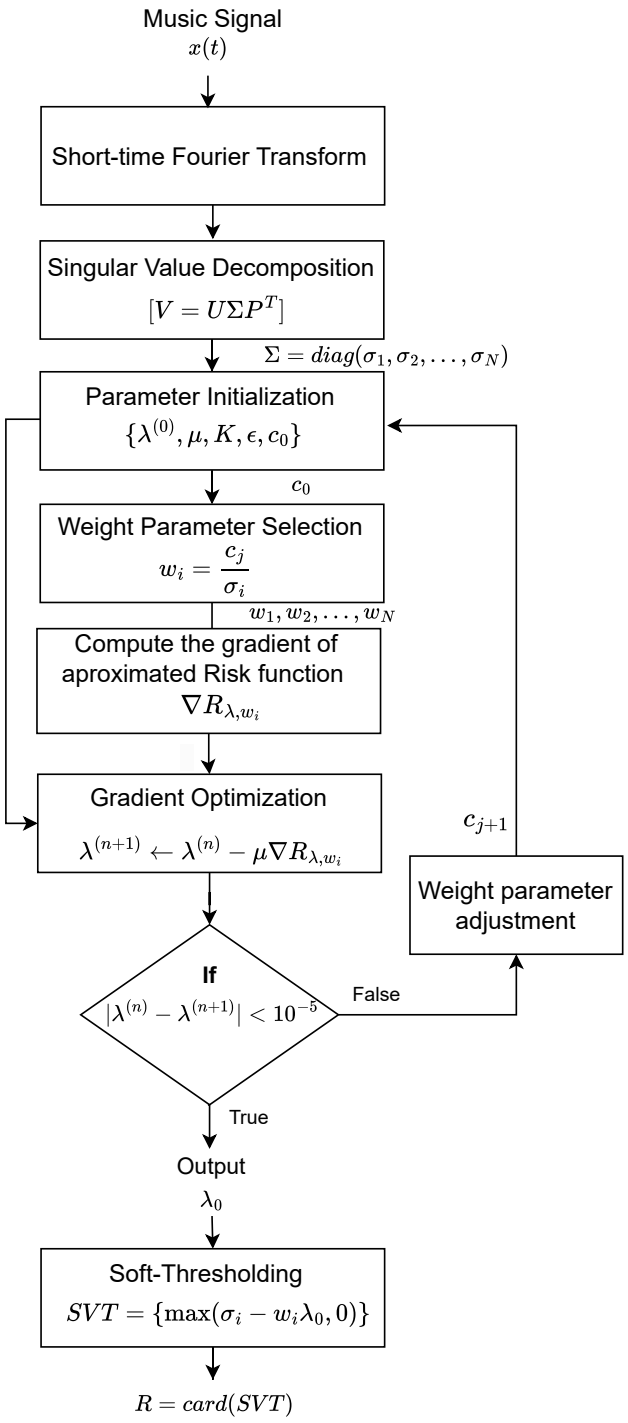

Figure 3.2: The flow chart for proposed GWSVT-SURE algorithm.

Chapter 4

Experimental Results

In this section, the simulation results of the implemented NMF algorithms as well as of the proposed algorithms are presented. Experiments for NMF algorithms are performed to compare the existing algorithms and identify any significant differences in terms of convergence rate and computational complexity. For the proposed algorithms, experiments are performed to evaluate the note estimation performance as well as the computational complexity. It also included comparison with the state-of-art algorithms to establish the benchmarking and underline the advancements brought in the field as well as to identify areas for improvement.

4.1 Experimental Results: Nonnegative Matrix Factorization

The first experiment is conducted on a randomly generated numerical data, the matrix of 500×100 with rank $K = 20$ is chosen to analyze the convergence rate of different MU-NMF algorithms. The resulting learning curves for different β and α parameters of divergence are demonstrated in Fig. 4.1. The α parameter is a variable of a Renyi divergence which incorporates Pearson-χ^2 ($\alpha = 2$), Neyman ($\alpha = -1$) , and Hellinger ($\alpha = -0.5$) divergences. The results shows that Renyi divergence fits best for random numerical data with Neyman divergence having the best convergence rate among compared algorithms. Furthermore, it is seen that Hellinger, KL, IS divergences has similar steady state performances, where the IS divergence is more robust compared to other algorithms. On the other hand, FN and Pearson-χ^2 divergences perform worst, having the same steady state performance. To sum up, results shows that the behavior of NMF is highly dependent on the choice of divergence and application at hand. Therefore, one must carefully investigate the performance of an algorithm for different types of divergences in order to achieve the optimal solution.

Fig. 4.2 represents the computational complexity of a MU-NMF algorithm with divergence value representing KL-divergence. Results show that algorithm

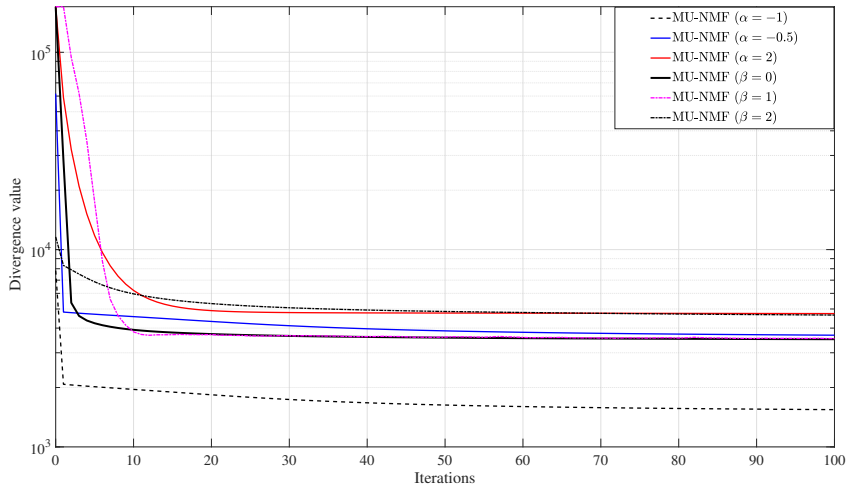

Figure 4.1: The learning curves of MU-NMF algorithms for different values of β and Renyi divergences.

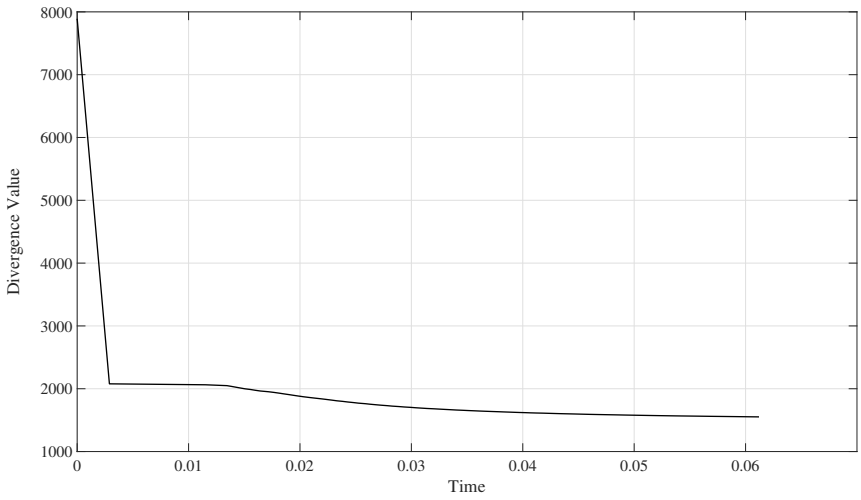

Figure 4.2: The convergence rate with respect to time for MU-NMF algorithm.

achieves fast convergence time of 0.06 seconds.

In this experiment, the random input matrix of same size and rank is generated (500×100, $K = 20$) to evaluate the performance of presented NMF algorithms. Fig. 4.3 and Fig. 4.4 demonstrates the learning curves and convergence time for four different NMF algorithms, respectively. The divergence value is set to be Euclidean

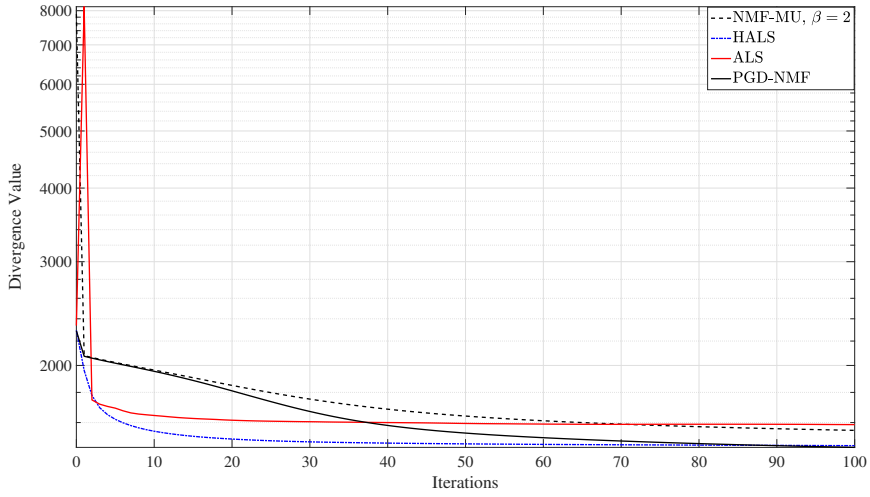

Figure 4.3: The learning curves behavior for different NMF algorithms.

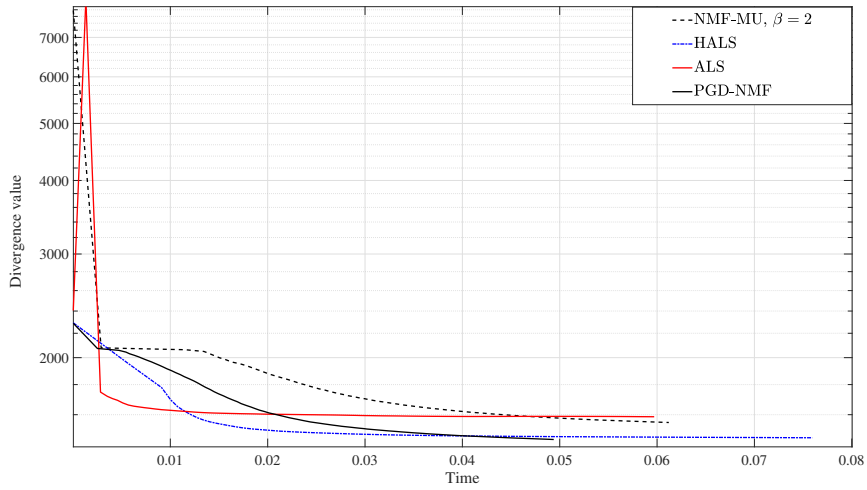

Figure 4.4: The convergence rate with respect to time for different NMF algorithms.

distance (Frobenius norm). Overall, it is apparent that HALS-NMF outperforms other algorithms in terms of robustness, convergence rate and computational complexity. On the other hand, MU-NMF with KL divergence has the worst performance for similar comparison metrics. Furthermore, the MU-NMF (KL) and ALS-NMF has similar steady state performance, although the latter is more robust and achieves faster convergence. The similar situation could be observed for PGD-NMF

and HALS-NMF. It is also important to note that NNLS based NMF algorithms, HALS-NMF and ALS-NMF shows faster convergence compared to multiplicative and gradient based optimization methods.

4.2 Experimental Results: Proposed Algorithms

In this section, the proposed WSVT-SURE, GWSVT-SURE algorithms are evaluated with piano music recordings within three different cases and compared with benchmark algorithms. Namely, for the first case, the proposed algorithms are tested on eight synthetic recordings with high polyphonic note components ranging from 9 to 42. Figure 4.5 represent the spectrogram corresponding to synthetic piano recordings. In the second case, the evaluation is conducted on twenty eight real piano music excerpts from the MIDI-Aligned Piano Sounds (MAPS) database (AkPnCGdD), with number of notes varying from 18 to 58 [50]. In the third case, the MAPS dataset is used to assess the performance of the proposed algorithms for the noisy recordings. Furthermore, experiments have been performed to investigate the computational complexity for 5 music signals with duration of 5 to 25 seconds. The STFT of recordings has been first computed with a Hanning window of length $L = 2048$ and 50 percent overlap after which a power spectrogram is computed and used for the subsequent experiments.

The proposed algorithm has been compared with conventional and state-of-the-art rank estimation algorithms including ARD-NMF [30], nPCA-SURE [32] and SVT-SURE [34]. The ARD-NMF variation with the l_2 norm has been chosen with parameters $a = 5$, $\beta = 0$, $\tau = 1.5 \times 10^{-5}$ resulting to the best estimation accuracy. For the SVT-SURE [34], nPCA-SURE [32] , and proposed algorithms, the noise variance is set to constant value of $\tau_e^2 = 1$. For the computational estimation of the noise variance, the first version of RESURE [33] algorithm has been chosen. Weights of the proposed WSVT-SURE algorithm are adjusted manually by varying the constant factor of c and s in (3.12). Furthermore, for GWSVT-SURE the accuracy controlling parameters K, ϵ are set as $K = 15$, $\epsilon = 0.01$, returning the best result for note estimation. The initial value λ^0 is set to $\lambda^0 = 1$ and the step-size parameter μ is found through empirical observations.

4.2.1 Parameter Tuning

For WSVT-SURE, the parameter c is set to a small value, chosen empirically from the set of smallest singular values. This choice is motivated by the fact that smaller singular values contribute to a higher weight parameter, leading to significant shrinkage and further annihilation by the soft thresholding operator. The choice of parameter c is critical for note estimation accuracy, as even slight variations can have a huge impact on the resultant estimation. Fig. 4.6 illustrates the behavior

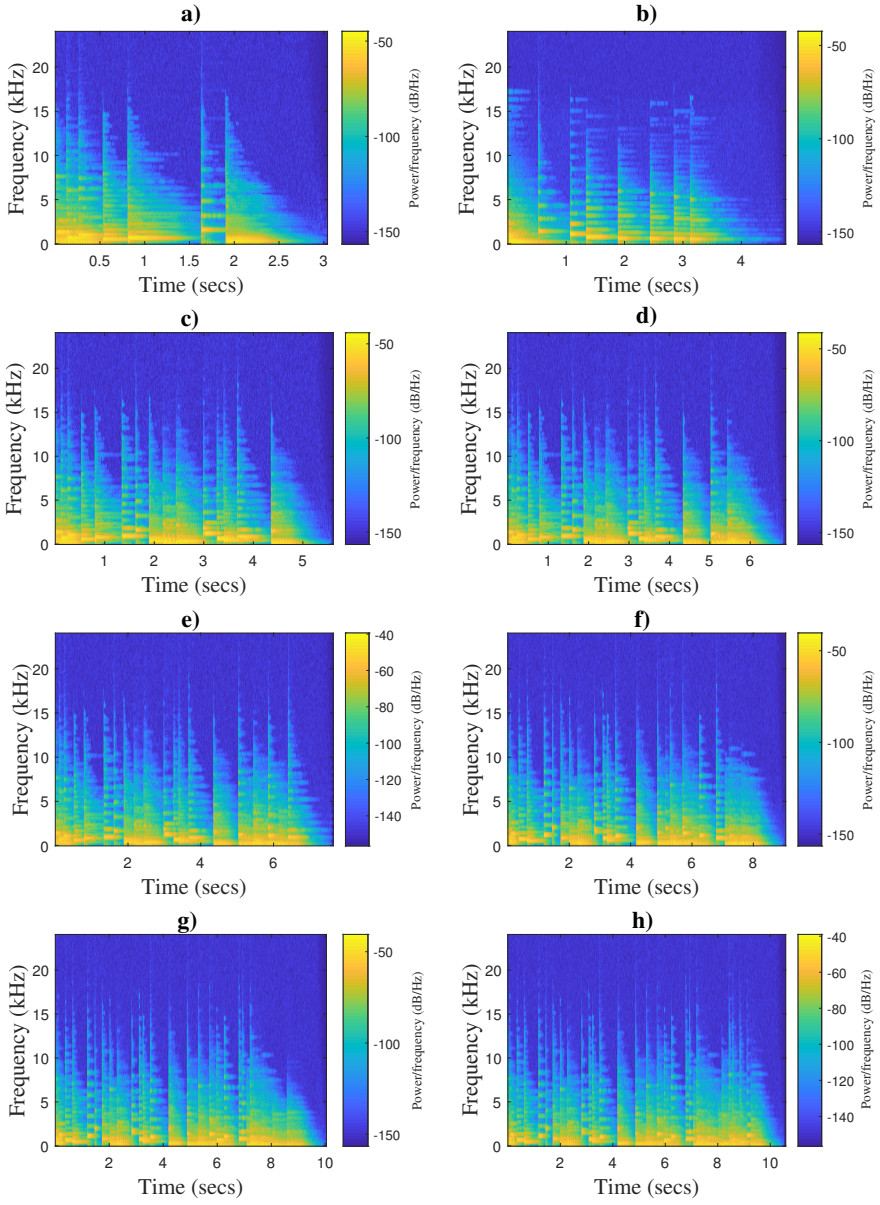

Figure 4.5: Time-Frequency representation of the synthetic piano music data, a)-h) expresses recordings with note number 9-42, respectively

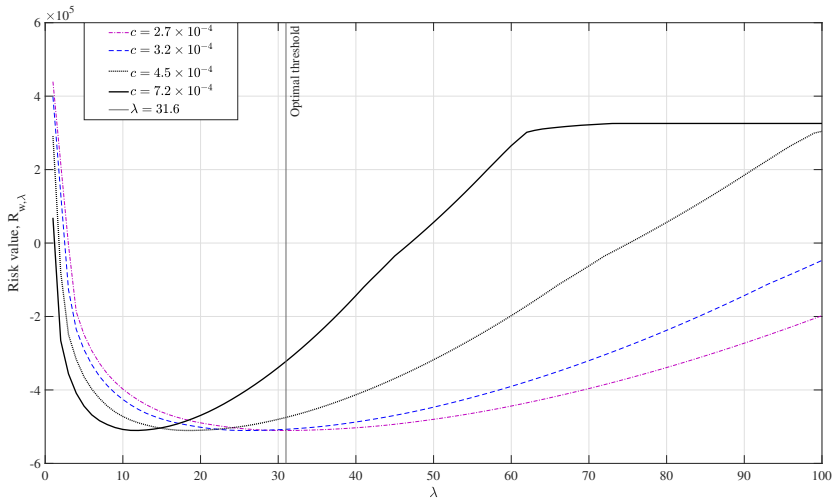

Figure 4.6: The behaviour of the risk function with respect to threshold values for several weight vector parameters.

of the risk function for 23 piano note recording with varying weight vector parameters c. The global minima of the risk function gradually converge toward the optimal threshold $\lambda_0 = 31.6$ (shown as a vertical line) as the c parameter is reduced. A view of weight vectors for synthetic data is provided in Fig. 4.7. This representation might give a false impression on an inverse relation between the optimal weight parameter c and the number of musical notes. Similarly, view of weight vector parameter on real data provided in Fig. 4.8 presents the completely random relation between note number and c parameter. Hence, the further analysis is required to identify any dependence of c parameter.

Parameter s is used to control the magnitude of the weighted threshold, in order to achieve the desired result for the soft-thresholding. This parameter has been introduced because effect from the c parameter alone typically results in excessively high or low values for the weighted thresholds. Therefore, the value of s parameter has been chosen according to the magnitude of the resulting $w_i\lambda$ and the singular values of the original music data. This parameter typically varies from -1 to 15 and chosen manually by first setting $s = 0$ and inspecting the produced $w_i\lambda$, in the case of over-shrinkage, this parameter is either set to small, negative value, or if the produced weighted thresholds are small, it is gradually increased until the correct rank is not reached.

Fig. 4.9 displays the learning curves of the GWSVT-SURE algorithm on a 23-note recording. It is apparent that proposed algorithm is sensitive to minor changes in the step-size. Detailed results indicate: 1) the upper bound of the optimal step-size depends on the number of musical notes in the data, and 2) an inverse relation

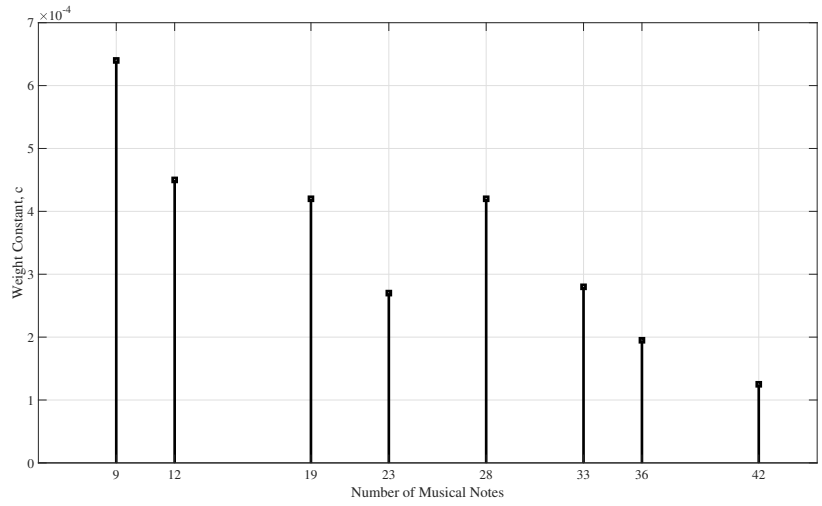

Figure 4.7: Optimal weights of the proposed WSVT-SURE algorithm for various number of musical notes.

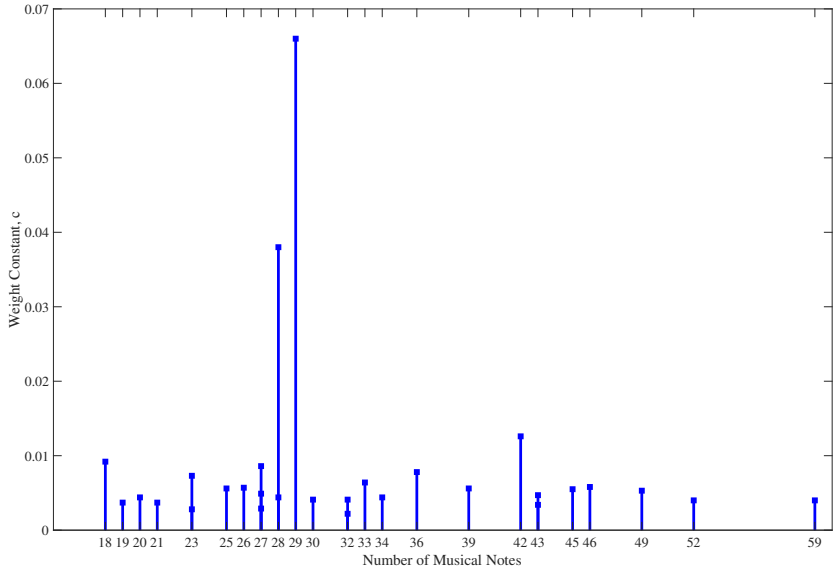

Figure 4.8: Optimal weight vector parameter of WSVT-SURE for real piano music

exists between the optimal step-size and the number of musical notes. Hence, adjusting the step-size parameter for each recording is necessary. Similarly, Fig. 4.10 illustrates λ thresholds determined by GWSVT-SURE for different musical note recordings. It is seen that, as the number of notes decreases, the estimated optimal threshold also decreases. Moreover, it is observed that a smaller value for the step-size parameter should be selected as the number of musical notes

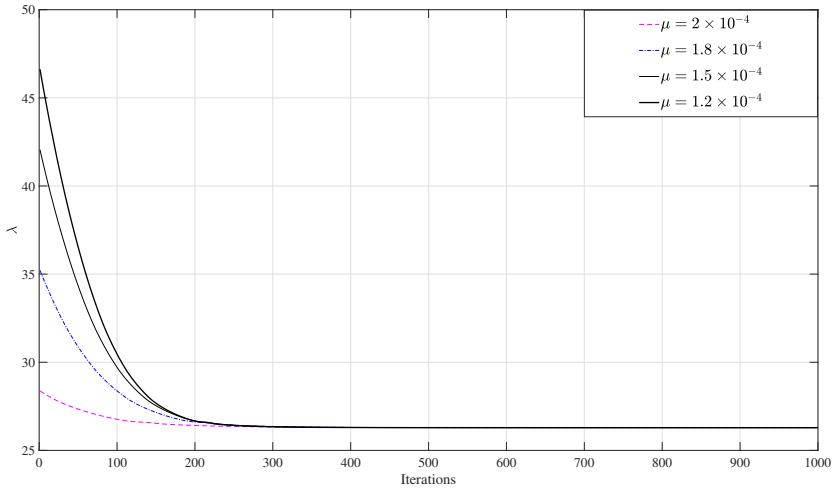

Figure 4.9: The effect of changing step-size on the convergence behaviour.

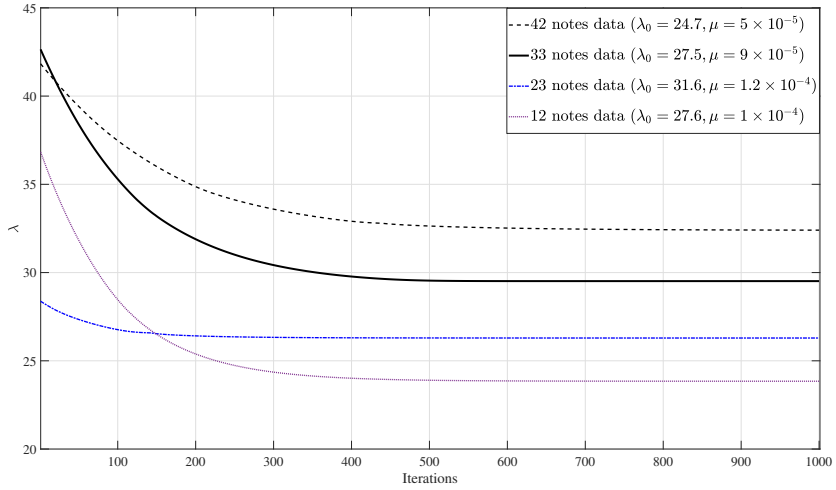

Figure 4.10: Gradient estimation of optimal thresholds for different recordings.

increases.

4.2.2 Case 1: Note Estimation Performance with Synthetic Data

Table 4.1 displays the accuracy of piano note estimation for various algorithms across eight music recording with different number of notes. Bold and underlined numbers are used to indicate the best and second-best results, respectively.

Table 4.1: Number of piano note estimation results for 8 synthetically generated music data

Num. of notes	9	12	19	23	28	33	36	42	MAE	MAD
ARD-NMF [30]	8	9	14	21	23	22	28	33	5.50	2.88
RESURE [33]	11	11	17	21	25	26	27	31	4.62	3.78
nPCA-SURE [32]	11	11	17	**22**	27	26	29	33	3.75	2.94
SVT-SURE [34]	11	**12**	23	27	32	**34**	37	39	2.38	1.38
proposed WSVT-SURE	**9**	**12**	**21**	**22**	**28**	**34**	**36**	**41**	**0.63**	**0.94**
proposed GWSVT-SURE	11	**12**	23	27	31	**32**	41	39	2.75	4.56

Accuracy is evaluated using Mean Absolute Error (MAE), being defined as:

$$\text{MAE} = \frac{1}{Q}\sum_{i=1}^{Q}|\hat{y}_i - y_i|, \tag{4.1}$$

where Q represents the number of test signals, and $\hat{y}$, y denote the true and estimated number of musical notes, respectively. The measure of dispersion is chosen as the Mean Absolute Deviation (MAD) from MAE:

$$\text{MAD} = \frac{1}{Q}\sum_{i=1}^{Q}|(\hat{y}_i - y_i) - \text{MAE}|, \tag{4.2}$$

where MAE represents the MAE.

Table 4.1 demonstrates that WSVT-SURE attained almost perfect estimation results among all of the estimators. Moreover, the estimation performance of the proposed WSVT-SURE and GWSVT-SURE algorithms remains stable with the change of number of notes in the recordings. In fact, both of the proposed algorithms achieve moderately lower dispersion rates compared to the benchmark algorithms. Additionally, comparison of SVT-SURE [34] with WSVT-SURE demonstrates that non-uniform shrinkage substantially improves the algorithm performance. Furthermore, it is observed that GWSVT-SURE indeed reaches the minima of a risk function, which is reflected in the form of almost comparable results with SVT-SURE [34]. Considering the RESURE [33] and nPCA-SURE [32] algorithms, the estimation performance is approximately equal, although the nPCA-SURE [32] has slightly better MAE and MAD. This might not seem reasonable because RESURE [33] implements noise variance estimation, which is believed to improve the performance compared to nPCA-SURE [32]. Also, results reveal that estimation performance of ARD-NMF [30] gradually decreases with the increase of number of notes in the recording. This explains the worst results for MAE but much better MAD index, outperforming both RESURE [33] and nPCA-SURE [32].

4.2.3 Case 2: Note Estimation Performance with Real Piano Recordings

Table 4.2 demonstrates the accuracy of note estimation of five algorithms evaluated with 28 real piano recordings from MAPS [50] dataset. The results show that proposed WSVT-SURE algorithm achieved the best result according to MAE and MAD compared to benchmark algorithms. Moreover, it also reached the best estimation performance at each of the test recording with maximum deflection of 1 note. Similarly, in comparison with benchmark algorithms, the proposed GWSVT-SURE is shown to have comparable results to the proposed WSVT-SURE. The performance of the proposed GWSVT-SURE is stable to the change of notes, with the errors spread uniformly across all of the recordings. On the other hand, ARD-NMF [30] is shown to have better estimation performance in recordings with small number of notes, and steadily worsened performance as the number of notes increases. It is also shown to overestimate the latent dimensionality, achieving the highest MAD value compared with other benchmark algorithms. The accuracy of SVT-SURE [34] algorithm for real recordings is significantly degraded when compared with results on synthetic data. Results show that algorithm often underestimates the true number of notes, leading to better performance in data with small number of notes. It is also apparent that both of the proposed algorithms considerably improve the performance of SVT-SURE [34] in terms of accuracy. For the nPCA-SURE [32], algorithm is ranked as poorest in terms of note prediction with major inaccuracies in estimation. Close inspection of the result reveal that algorithm is unable to track the information regarding the note number in the data, leading to the underestimation and high deflection rates.

Table 4.2: Number of piano note estimation results of four algorithms for 28 real piano recordings

Num. of notes	28	30	34	27	33	42	27	58	52	23	25	32	26	32	20	43	49	46	19	39	21	23	27	28	29	45	36	18	MAE	MAD
ARD-NMF [30]	11	24	26	30	15	56	44	39	66	20	54	29	36	**31**	65	61	17	21	30	44	24	24	31	42	32	53	31	29	12.3	18.5
nPCA-SURE [32]	8	12	10	4	8	11	16	11	3	12	9	7	15	14	19	17	10	15	5	12	5	16	8	7	14	9	14	11	21.8	8.7
SVT-SURE [34]	13	17	12	6	11	7	18	36	29	13	17	12	6	11	7	18	36	29	24	25	33	25	32	20	44	51	45	4	14.9	9.8
proposed WSVT-SURE	**28**	**30**	**34**	**27**	**32**	**41**	**27**	**58**	**52**	**24**	**25**	**32**	**27**	**31**	**20**	**43**	**49**	**46**	**19**	**39**	**21**	**23**	**27**	**27**	**29**	**45**	**35**	**17**	**0.3**	**0.5**
proposed GWSVT-SURE	32	34	**34**	28	28	45	30	53	59	27	29	39	30	35	18	51	48	51	21	34	29	27	30	33	33	50	38	14	4	6.7

Table 4.3: Number of piano note estimation results of three algorithms for 8 noisy piano recordings

	Notes	21	26	28	30	34	36	42	MAE
Noiseless	SVT-SURE [34]	28	6	20	17	12	45	7	19
	proposed WSVT-SURE	20	26	28	30	34	36	40	0.1
	proposed GWSVT-SURE	20	26	28	30	34	36	42	0.9
40 SNR	SVT-SURE [34]	4	8	15	17	13	11	10	19.8
	proposed WSVT-SURE	**21**	**26**	**28**	**30**	**34**	34	52	1.71
	proposed GWSVT-SURE	19	25	**28**	28	**34**	**36**	49	1.71
30 SNR	SVT-SURE [34]	9	9	17	19	15	12	10	18.0
	proposed WSVT-SURE	20	**26**	24	28	35	38	38	1.57
	proposed GWSVT-SURE	20	**26**	24	28	32	34	48	2.42
20 SNR	SVT-SURE [34]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	proposed WSVT-SURE	20	26	28	30	34	36	40	0.1
	proposed GWSVT-SURE	19	25	26	28	32	**36**	48	2.14

4.2.4 Case 3: Note Estimation Performance with Noisy Recordings

This experiment is motivated by the fact that musical data is often transferred by communication channels which distorts the information in the form of noise. Therefore, AMT algorithms must not lose its functionalities, being robust to noises present in the data. Table 4.3 represents the accuracy of note estimation in the case of 8 noisy recordings for proposed WSVT-SURE, GWSVT-SURE algorithms compared with SVT-SURE [34] baseline algorithm. For this experiment, Additive White Gaussian Noise (AWGN) of power of 40 SNR to 20 SNR is chosen. This choice is based on the perceptual quality of the noisy signal being used for the experiments.

Results show that proposed WSVT-SURE algorithm achieve the best performance in terms of MAE in all of the considered noise power scenarios. It is seen that algorithm has lost only around 1.5 of MAE even in the presence of strong noise source. Similarly, proposed GWSVT-SURE algorithm achieved comparable results with WSVT-SURE algorithm with slight increase of MAE at 30 SNR and 20 SNR noise. It is generally apparent that high portion of error corresponding to proposed GWSVT-SURE algorithm comes from incorrect estimation of 42 note recording, whereas in recordings containing less notes (21-36), algorithm perform at almost same level as proposed WSVT-SURE algorithm. Considering SVT-SURE [34], algorithm struggles to correctly estimate the number of musical notes in all of the considered noise powers. Moreover, at 20 SNR noise, algorithm diverges to infinity whereas in other scenarios it shows increase of almost 5 values of MAE, compared to that of 1.5 points for the proposed WSVT-SURE, GWSVT-SURE algorithms. Overall, it could be concluded that proposed algorithms exhibit stability

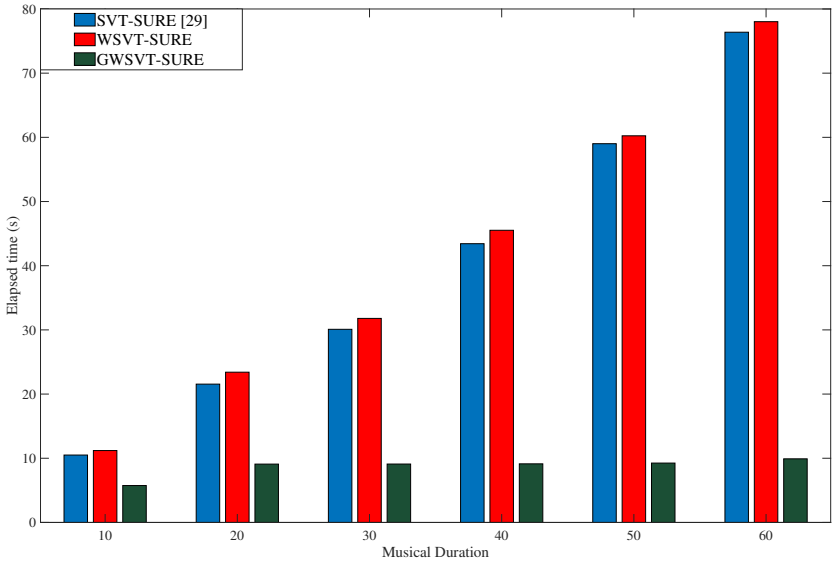

Figure 4.11: Elapsed time of three algorithms for different durations.

Table 4.4: Comparison of elapsed time (in seconds) of four algorithms with 5 music excerpts of different duration

Signal Duration	5	10	15	20	25
ARD-NMF [30]	1.144	3.647	5.283	9.627	10.721
RESURE [33]	3.141	6.482	8.921	11.134	13.111
nPCA-SURE [32]	**0.097**	**0.101**	**0.106**	**0.110**	**0.117**
SVT-SURE [34]	3.74	11.372	17.238	20.336	26.325
proposed WSVT-SURE	4.61	12.565	18.001	21.583	27.021
proposed GWSVT-SURE	0.51	1.85	2.01	2.31	2.54

for the strong presence of noise, having only minor increases in MAE.

4.2.5 Computational Complexity Analysis

Table 4.4 demonstrates the computational complexity analysis in terms of elapsed time (in seconds) of four algorithms for 5 music excerpts ranging from 5 to 25 seconds. The computational time is with reference to a desktop PC running MATLAB version 2018b on 8-core, Intel Core i9 vPro 9th Gen CPU with 16GB RAM.

SVT-SURE [34], WSVT-SURE rely on iteration through λ parameters which are chosen to be equal to 100 for all algorithms. The stopping criteria for GWSVT-SURE is set as $|\lambda^{n+1} - \lambda^n| < 10^{-5}$. Overall, the results show that nPCA-SURE [32]

algorithm is superior in terms of speed and stability for dimensionality change. Both SVT-SURE [34] and the proposed WSVT-SURE exhibit an increased computational complexity as compared with other benchmark algorithms. The proposed GWSVT-SURE algorithm is shown to have significantly improved computational complexity as compared with that of the SVT-SURE [34] and the proposed WSVT-SURE. The computational complexity of RESURE algorithm [33] is higher than that of nPCA-SURE [32] due to the fact that there is an additional step of noise variance estimation, requiring the use of onset frame estimation.

Figure 4.9 represents the comparison of SVT-SURE [34] with proposed WSVT-SURE, GWSVT-SURE algorithms for an extended music duration up to 1 minute. It is apparent that both SVT-SURE [34] and WSVT-SURE are costly in terms of computational complexity and unstable for the increase of dimensionality. However, it is worth noticing that the computational complexity of the proposed WSVT-SURE algorithm is comparable with that of the benchmark SVT-SURE [34], yet it outperforms SVT-SURE as well as the rest of the algorithms considered in this paper. Also, it is noticeable that proposed GWSVT-SURE is stable for growth of dimensionality and relatively fast compared to benchmark algorithms. Also, the proposed GWSVT-SURE provides the best trade-off between the estimation accuracy and computational complexity among the considered algorithms.

Chapter 5

Conclusion

5.1 Summary of Work done

This study addresses the issue of accurately and efficiently estimating musical notes in piano recordings which is essential for developing better NMF based AMT systems. The key idea of this study is to convert the real world problem of piano note estimation to the mathematical problem of estimating the rank of magnitude spectrogram matrix. To accomplish this, first, different variations of NMF algorithms are studied and implemented. This is followed by considering the rank reduction models such as SVT and model selection methods as SURE.

The primary directions of research included a) increasing the accuracy of note estimation and b) decreasing the computational complexity of a benchmark algorithm. The idea behind the proposed algorithms is to utlize the weighted variant of SVT-SURE risk function for better estimation performance and design a gradient based optimization algorithm to mitigate the high computational complexity.

Experimental results show that proposed WSVT-SURE achieves almost perfect accuracy in estimating the number of musical notes via manual weight adjustment. On the other hand, proposed GWSVT-SURE achieves substantial results in both note estimation and computational complexity, having a good trade-off between this parameters. The results of this Capstone Project has been submitted to the 2024 IEEE PACRIM Conference. Also, this work is currently being prepared for publication in IEEE Access journal. The complete information on conferences and journal publications could be found in List of Publications section.

5.2 Future Work

The key limitation of the proposed algorithm relies on manual weight adjustment for WSVT-SURE. Although, this work must be regarded as a proof of concept to demonstrate the capabilities of the proposed algorithm, the adjusted parameters

have a huge influence on the results of the experiments. Therefore, future research will be directed towards the derivation of a closed form expression for weight vector for AMT framework. Another important direction of future work would be to envisage strategies to analyse the stability of a GWSVT-SURE algorithm for various step-sizes as well as to develop its variable step-size modification. Finally, as the proposed algorithms at its essence a pre-processing stage, they must be combined in tandem with NMF algorithm in an efficient manner. This would fill the huge gap in the research field of NMF by offering a framework for rank estimation that combines flexibility, accuracy, and efficiency.

List of Publications

1. B. Kurmangaliyev and M. T. Akhtar, "Weighted Singular Value Thresholding and Gradient Optimization of Stein's Unbiased Risk Estimate with weight vector adaptation based on Akakike Information Criterion - Singular Value Decomposition (AIC-SVD)," in *IEEE Access*, 2024. [In Preparation]

2. B. Kurmangaliyev and M. T. Akhtar, "Weighted Singular Value Thresholding and Gradient Optimization of Unbiased Risk Estimate for Rank Estimation in Automatic Music Transcription," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'24)*, August 21-23, 2024, Victoria, B.C., Canada [SUBMITTED]

Bibliography

[1] J.Y. Wang and J.-S. R. Jang, "Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 383–396, 2023.

[2] S. Yong, L. Su, and J. Nam, "A phoneme-informed neural network model for note-level singing transcription," in *Proc. IEEE ICASSP*, 2023.

[3] M. Heydari, J.-C. Wang, and Z. Duan, "SingNet: A real-time singing voice beat and Downbeat Tracking System," in *Proc. IEEE ICASSP*, 2023, pp. 6112–6118.

[4] J. Pauwels and G. Peeters, "Evaluating automatically estimated chord sequences," in *Proc. IEEE ICASSP*, 2013.

[5] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *Proc. IEEE ICASSP*, 2019

[6] M. Krause and M. Müller, "Hierarchical classification for instrument activity detection in orchestral music recordings," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2567–2578, 2023.

[7] P. Yu and H. Chen, "Deep multilevel cascade residual recurrent framework (MCRR) for sheet music recognition," *IEEE Access*, vol. 12, pp. 6941–6960, 2024.

[8] Tao Li and M. Ogihara, "Music genre classification with taxonomy,"in *Proc. IEEE ICASSP*, 2005, pp. 197–200.

[9] Y. Li, H. Liu, Q. Jin, M. Cai, and P. Li, "TROMR:transformer-based Polyphonic Optical Music recognition," in *Proc. IEEE ICASSP*, 2023.

[10] Y.-T. Wu, B. Chen, and L. Su, "Polyphonic Music Transcription with Semantic Segmentation," *Proc. IEEE ICASSP*, 2019.

[11] A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, Jun. 2000.

[12] Hyvarinen Aapo, J. Karhunen, and E. Oja,*Independent Component Analysis*. New York: Wiley, 2001.

[13] Lee. S, "Estimation of the Matrix Rank of Harmonic Components of a Spectrogram in a Piano Music Signal", *IEICE Tran. Information and Systems*, vol. E102.D, no. 11, pp. 2276-2279, 2019

[14] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for Polyphonic Music transcription," in *Proc. 2003 IEEE Workshop Applications Signal Processing Audio Acoustics*, 2003, pp. 177-180.

[15] C. Fevotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4810–4819, 2015.

[16] B. Schuller, F. Weninger, M. Wollmer, Y. Sun, and G. Rigoll, "Non-negative matrix factorization as noise-robust feature extractor for speech recognition," in *Proc. IEEE ICASSP*, 2010, pp. 4562-4565.

[17] J.-W. Hung, H.-J. Hsieh, and B. Chen, "Robust speech recognition via enhancing the complex-valued acoustic spectrum in modulation domain," *IEEE/ACM Trans. Audio Speech Language Processing*, vol. 24, no. 2, pp. 236–251, 2016.

[18] N. Kumar et al., "Hyperspectral tissue image segmentation using semi-supervised NMF and hierarchical clustering," *IEEE Trans. Medical Imaging*, vol. 38, no. 5, pp. 1304–1313, 2019.

[19] D. Guillamet, B. Schiele, and J. Vitria, "Analyzing non-negative matrix factorization for Image Classification," in *Proc. IEEE ICPR*, 2002, pp. 116-119.

[20] C. Fevotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[21] Z. Yang and E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction", *Proc. of Scandinavian conference on Image Analysi*s, pp. 333-342, 2005.

[22] Kannan, G. Ballard, and H. Park, "MPI-Faun: An MPI-based frame work for alternating-updating nonnegative matrix factorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 544–558, 2018.

[23] A. Cichocki, R. Zdunek, and S.-ichi Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," *Independent Component Analysis and Signal Separation*, pp. 169–176, 2012

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

[25] S. Squires, A. Prugel-Bennett, and M. Niranjan, "Rank selection in non negative matrix factorization using minimum description length," N*Neural Computation*, vol. 29, no. 8, pp. 2164–2176, 2017

[26] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004

[27] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 4, pp. 766–778, 2008.

[28] S. Kırbız and B. Günsel, "Perceptually Enhanced Blind Single-channel music source separation by non-negative matrix factorization," *Digital Signal Processing*, vol. 23, no. 2, pp. 646–658, 2013.

[29] L. Li, "Model selection via minimum description length," thesis, Graduate Department of Statistics University of Toronto, Toronto, 2011.

[30] Y. Tan and C. Fevotte, "Automatic relevance determination in nonnegative matrix factorization with the /SPL beta/-divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1592–1605, 2013

[31] L. Muzzarelli, S. Weis, S. B. Eickhoff, and K. R. Patil, "Rank selection in nonnegative matrix factorization: Systematic comparison and a new mad metric," *2019 International Joint Conference on Neural Networks (IJCNN),* 2019

[32] M. O. Ulfarsson and V. Solo, "Rank selection in noisy PCA with SURE and Random Matrix Theory," in *Proc. IEEE ICASSP*, 2008, pp. 5804-5816.

[33] S. Lee, "Estimating the rank of a nonnegative matrix factorization model for automatic music transcription based on Stein's unbiased risk estimator," *Applied Sciences*, vol. 10, no. 8, p. 2911, 2020.

[34] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko, "Unbiased risk estimates for singular value thresholding and spectral estimators," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4643–4657, 2013.

[35] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré, "Stein Unbiased Gradient Estimator of the Risk (SUGAR) for Multiple Parameter Selection," *SIAM Journal Imaging Sciences*, vol. 7, no. 4, p. 2448, 2014

[36] S. Gu et al., "Weighted nuclear norm minimization and its applications to low level vision," in *International Journal Computer Vision*, vol. 121, no. 2, pp. 183–208, 2016.

[37] N. Gillis, *Nonnegative Matrix Factorization*. Philadelphia: Society for Industrial and Applied Mathematics (SIAM), 2021.

[38] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," *Proc. IEEE ICASSP*, 2014, pp. 6201–6205.

[39] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. New York: Wiley, 2009.

[40] A. Cichocki and R. Zdunek, "Regularized alternating least squares algorithms for non-negative matrix/tensor factorization," *Advances in Neural Networks - ISNN 2007*, pp. 793–802, 2007.

[41] A. H. Phan, A. Cichocki, K. Matsuoka, and J. Cao, "Novel hierarchical ALS algorithm for nonnegative tensor factorization," *Proc. IEEE ICASSP*, 2011, pp. 1984–1987

[42] H. Trussell and M. Civanlar, "The Landweber iteration and projection onto convex sets," *IEEE Trans. Audio Speech Lang. Process.*, vol. 33, no. 6, pp. 1632–1634, Dec. 1985.

[43] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, UK , 1998.

[44] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[45] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[46] P. Nobel, E. Candès, and S. Boyd, "Tractable evaluation of Stein's unbiased risk estimate with convex regularizes," *IEEE Trans. Signal Process.*, vol. 71, pp. 4330–4341, 2023.

[47] J.-F. Cai, E. J. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[48] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, 1981.

[49] K. Biswas, S. Kumar, S. Banerjee, and A. K. Pandey, "Smooth maximum unit: Smooth activation function for deep networks using smoothing maximum technique," *Proc. IEEE/CVF CVPR*, pp. 794-803, 2022.

[50] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1643–1654.