BUILDING AN NMF SOURCE SEPARATION TOOLBOX FOR MUSICAL AUDIO

MIDTERM REPORT

MATTY BOI CCCCCC

# Abstract

Non-Negative Matrix Factorization (NMF) has proven to be an effective tool in source separation problems for musical audio. This report presents a MATLAB framework for source separation using NMF. Several related algorithms have been implemented and benchmarked, and the software is highly modular and extensible. We also present a discussion and timeline of future work, including score-aware implementations and public release as a MATLAB toolbox.

# Contents

# Introduction

*Source Separation* is the name given to the problem of extracting a set of individual sound *sources* from one or several *mixtures*, where a mixture is a weighted sum of the sources whose weighting may change with time. Mixtures may be *Instantaneous* – affected only by the present values of the sources – or *convolutive* – affected by present and past values. In some cases, the problem can be exactly solved, in theory allowing perfect reconstruction of the source signals. There are inherent ambiguities, however, in *underdetermined* mixtures with more sources than mixture channels. Algorithms for this class of problems must make prior assumptions about the source signals, such as statistical independence, harmonicity, or sparseness under some frequency transform.

Applications of source separation algorithms are numerous and include noise reduction, speech enhancement and analysis of hyperspectral images [1]. In music, source separation can be used for *upmixing* mono to stereo or stereo to surround sound, and for *remastering* existing recordings – perhaps by extracting the sound of a single instrument, editing it, and replacing it in the mix. More exotic uses include automatic generation of karaoke backing tracks [2]. These applications often involve underdetermined mixtures while requiring high quality reconstruction, placing heavy demands on the source separation algorithm. Musical Source Separation is an area of active research.

This project aims to create a robust MATLAB framework for Source Separation of musical audio using Non-Negative Matrix Factorization (NMF), a technique which approximates the Short Time Fourier Transform (STFT) matrix of a mixture as a product of two non-negative matrices of much smaller rank. There are a range of possible NMF-based algorithms depending on the choice of approximation cost function and the application of various constraints. *Score-aware* approaches which incorporate information from a musical score will be a particular focus.

The goals of the project are as follows:

1. Produce a flexible software framework for source separation using NMF which can accommodate a wide range of algorithms
2. Produce a framework which is modular and trivially easy to extend, and therefore useful for other researchers
3. Implement a range of blind and score-aware source separation algorithms, and compare their performance using established benchmarks.
4. Distribute the project codebase online for free as a MATLAB toolbox

For these goals to be met, the code must be production quality throughout the codebase, with effective error handling, extensive documentation and commenting, and minimal coupling between modules. High performance, though desirable in the end product, is a secondary concern at present. Reproducibility of results is ensured by versioning and publicly releasing code, using publicly available datasets, and benchmarking with standard benchmarks. A detailed project plan and logbook have been maintained throughout and will be taken forward into the second half of the project.

This Report is structured as follows:

<blah blah>

# Literature Review

Cover interpretations of nmf quantities

# Work To Date

## Overview

Work on the project to date has consisted of background research, design and architecture tasks, implementation, and testing. A generic framework for source separation has been fully architected and implemented in a GitLab repository along with several blind NMF-based algorithms and a series of test scripts. Score alignment and score-aware source separation algorithms have yet to be implemented, though their place in the architecture has been carefully mapped out.

Test data comes from the TRIOS [3] and PHENICX [4], [5], [6] HOW CITE source separation datasets, which also include scores and score alignment information for the next phase of the project. Testing included full pipeline benchmarking in several configurations, as well as a more targeted look at ISTFT reconstruction quality. Fuzz testing was used to assess the robustness of the NMF algorithms. When assessing the whole pipeline, two preexisting benchmarks from the literature were used - BSS\_EVAL [7] and PEASS [8].

## Framework design

### Choice of Language

The first task in architecting the system was to choose a language. MATLAB, Python with Numpy, and lower-level approaches including C and C++ were considered. C was ruled out due to its error prone nature and lack of portability. C++ fares a little better on these two counts but lacks native matrix operations. MATLAB and Python are both feasible contenders from a technical point of view – MATLAB was chosen for its widespread adoption and integrated debugging tools despite Python’s more expressive syntax.

### The Generic Source separation algorithm

Reiterate interpretations of nmf values. Therefore source sep happens in four parts – spect, set init matrices, converge, reconstruct. describe steps required to reconstruct phases. Mention inherent coupling between stft and istft

Diag – source sep algo.

Needed a highly generic and extensible format but with structure. So pass functions around! “source sep” algo just calls them in turn and combines the results. V v generic but ensures common bare-minimum interface (even though interface of func itself can change!). If args need passing can use function partials and @ notation. Explain what a function partial is but leave til impl to show how it works. Managing args since no named args. Making pipeline reconfigurable. Describe interfaces. Programmers responsibility to ensure sensible functions.

Diag – sep\_sources

### Proposed score aware source separation architecture

Most NMF algorithms work by constraining W\_init, H\_init – can get v far on that alone. Eg score align by passing audio and score to nmf\_init\_aligned, which calls out to a score alignment function.

Diag – proposed score aligned source separation architecture

Score alignment using DTW will be architected as follows

Diag – how to score align using a dtw algorithm

## Implementation - Blind Source Separation

### Repository Structure

Diag – repo folder struct

This corresponds to an architecture of

Diag – arch

### nmf\_separate\_sources

### NMF functions

# future work

# summary

# Conclusions

# references