Modulo7 : A full stack Music Information Retrieval and Querying Engine using Music Theoretic Principles

by

Arunav Sanyal

A thesis submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science.

Baltimore, Maryland

December, 2015

Abstract

Music Information Retrieval (MIR) is an interdisciplinary science of extracting non trivial information and statistics from music data sources. In today's digital age, music is stored in a variety of digitized formats - e.g midi, musicxml, mp3, digitized sheet music etc. Music Information Retrieval Software aim at extracting features from one or more of these source. MIR research helps in solving problems like automatic music classification, recommendation engine design etc. Users can then query the acquired statistics to acquire relevant information.

The author proposes and implements a new Music Information Retrieval and Query Engine called Modulo7. Unlike other MIR software which deal with low level audio features, Modulo7 operates on the principles of music theory and a symbolic representation of music. Modulo7 is a full stack deployment, with server components that parse various sources of music data into its own efficient internal representation and a client component that allows consumers to query the system with sql like queries which satisfies certain music theory criteria (and as a consequence Modulo7 has a

ABSTRACT

custom relational algebra with its basic building blocks based on music theory).

Primary Reader: Dr David Yarowsky

Secondary Reader: Dr Yanif Ahmad

Acknowledgments

I would like to thank Dr David Yarowsky for giving me the opportunity to work on this project. His detailed insights have immensely helpful to me to power through my work. I would like to thank Dr Yanif Ahmad for his crucial help in the systems aspects of my query engine and the implementation of the server side components.

Dedication

This thesis is dedicated to my family and to all the music lovers in the world.

Contents

Chapter 1

Introduction

Why does a person like a particular song? What are the inherent aspects of a song that pleases a person's musical taste? Is it the complexity of a song, the beat the song or just a particular melodic pattern ? More so if a person likes a song, can we predict if he/she will like a similar song?

Music has been created since the dawn of civilization and these questions have plagued mankind just as long. In response to this, man has created elaborate systems of formal study for music and classification techniques in almost every ethnic community since antiquity. Two notable examples are the western system of solfege and classical music theory and the Indian system of raagas. These elaborate systems are based on very simple fundamental building blocks of melody and harmony and simple rules that govern the interplay of these building blocks. However very complex pieces of music can be created with these simple rules depending on the skill and virtuosity of artists. Composers use these rules and concepts to create novel music for mass consumption.

In the modern era industry and academia have attempted to address the problem of music recommendation and music classification. The industry has predominantly favored approaches that look at user preferences and history. For example Amazon Music recommendation works on users shopping history. Pandora on the other hand hires an army of musicologists to ascertain how a song is similar to another song and creates software that leverages this adhoc generated data. These approaches are either expensive in the human labor needed or in the amount of data processed that is input from a large number of users. More recently, companies like Echo Nest has extensively extracted features from music sources and mined cultural information on the web but leave it at consumers how best to leverage the data. Hence symbolic MIR is not traditionally used in industry and music theory is an after thought.

Academia on the other hand attempts to solve very particular problems in MIR. Typical examples would be cover song detection, processing information via signal processing, audio feature extraction, optical music recognition etc. In most cases the applications are of a very specific domain and does not fully scale with bulk music data. Generic frameworks like the jMIR (which also happens to be a major inspiration for modulo7) suite for automatic music classification exists, which is meant to facilitate research in MIR with a machine learning focus. However academia is disconnected with industry and no full scale MIR engines can satisfy the scale of industry applications.

This work is attempt to bridge both communities. Modulo7 is a full stack deployment of Music Information Retrieval Software, providing both a server architecture and a sql like client to query based on music theory criteria. Modulo7 is a big data and information retrieval framework to explore the possibilities of exploring music theoretical aspects of music sources. Modulo7 does not attempt to solve very complex music theoretic problems (e.g study orchestral music to identify counter point information). Rather Modulo7 acts a framework on which such analysis can be built upon. Most importantly, Modulo7 addresses the issue of scale and allows a fast comparison between songs on certain music theoretic criteria. Modulo7 also acts to address deficiencies in existing software, such as filling up incompleteness in music sources. Certain problem statement of this sort would be Key estimation, Tempo estimation etc.

Chapter 2

Literature Review

Music Information Retrieval is an active and vibrant community.

Chapter 3

Basics of Music Theory

Music theory is defined as the systematic study of the structure, complexity and possibilties of what can be expressed musically. More formally its the academic discipline of studying the basic building blocks of music and the interplay of these blocks to produce complex scores (pieces of music). Modulo7 is built on top of western theoretic principles and hence only western music theory is explored. Also music theory is an extremely complicated subject and hence only the basics and relevant portions to the modulo7 implementation are discussed here.

Traditionally music theory is used for providing directives to a performer to play a particular song/score.

This section is primarily meant for people with a weak or lack of understanding of music theory. The following section talks about the basic building blocks of music theory:-

3.1 Building Blocks

Music is built on fundamental quantities (much like matter is built on fundamental quantities like atoms and molecules). The following are the core concepts in order of atomicity (i.e successive blocks build on the preceding ones)

3.1.1 Pitch/Note

A pitch is a deterministic frequency of sound played by a musical voice (instrument or human). In western music theory, certain deterministic pitches are encoded as Notes. For example the note A4 is equal to 440 Hz. In other words Notes are symbolic representations of certain pitches. With certain notable exceptions, most music is played on these set frequencies.

Each note is characterized by two entities. First is the note type and the second is the octave. An octave can be considered as a range of 12 notes. There are 8 octaves numbered 0 to 7 which are played by traditional instruments or vocal ranges. Then is the note type. Notes are categorized into 7 major notes (called A, B, C, D, E, F, G) and 5 minor notes (also called as accidentals). They can be characterized by increasing or decreasing the frequency of the notes by a certain amount (called sharps(#) and flats(b) respectively). For example the accidental lying in between (A and B is called A# or Bb). Similarly accidentals lie in between C, D; D, E; F, G and G, A. (Note that there are no accidentals in between B and C and E and F).

3.1.2 Semitone and Tone

A semitone is an increment or a decrement between two notes. For instance there is one semitone in between A and A#. Similarly there are 3 semitones in between A and C. A tone is an increment in between two major notes. Another characterization of a tone is two semitones.

Chord

A chord is a set of notes being stacked together (being played together at or almost at the same time). Chords are the basic building blocks of a concept called harmony (which will be discussed further on.). Traditionally a chord is constructed by stacking together notes played on a single instrument, but a chord can be constructed by different instruments simultaneously playing different notes.

3.1.3 Rests

Rests are pauses in between notes (with no sound being played at that point of time) for a fixed duration.

3.1.4 Melody

A melody is a succession of notes and rests which sound pleasing. There are many rules about what makes a melody sound good which we will get to in the subsequent reading.

3.1.5 Harmony

A harmony is a succession of chords (also known as a chord progression).

3.1.6 Voice

A voice is an interplay of notes, chords and stops by a single instrument/vocalist. The reader can think of a voice as a hybrid or generalization of the melody and harmony concepts.

3.1.7 Score/Song

A score or a song is an interplay of voices. It is the final product of music that is delivered to the audience. Songs are of different types based on cultural context and complexity (for example an orchestra is a large number of voices being coordinated by a conductor. In contrast a folk song might be played by a single person on a guitar or a duet between a vocalist and an instrumentalist).

Chapter 4

Software architecture

The following sections present the software architecture of Modulo7.

4.1 Server Side architecture

Modulo7 is designed with the purpose of scalability. A block diagram of the components of the server side architecture is presented below :-

1. Source Converter : Converts music sources (e.g. music XML, midi etc) into modulo7's binary representation.

2. Music Theory Models : The model is a description of music theoretic criteria that can be applied on top of a song. Examples would be melodic contour, tonal histogram etc.

3. Distributed Storage Mechanism : The modulo7 internal representation is a

conversion to create a song representation with all the meta data of the song (Key, Scale, etc) along with the sequences of note events stored as lists. This representation is then serialized and stored in and Hadoop Distributed File System. This allows for fault tolerance and a distributed deployment of the input data.

4. Lyrics Indexer : A distributed index of songs lyrics. This acts as a base on which standard techniques for similarity analysis might be applied. Alternatively it can provide a framework on which custom models (e.g. semantic intent of the song, correlation between music theory models and lyrics might also be applied).

5. Lyrics similarity models : A set of similarity models that can be applied to an index.

6. Query Engine : An SQL like interface to a client that allows you to gather and ascertain useful information (based on music theoretic criteria).

4.2 Client architecture

The server exposes a sql like interface as well as a consumable API. Some sample queries would be :-

1. select midi files from database where $melodic\_complexity > somethreshold$

2. select * from database where $artist = led\_zepplin\ and\ harmonic\_movement > harmonic\_movement(stairway\_to\_heaven)$

3. select $num\_voices\ from\ Database\ where\ songName = someSong.midi$

An API will also be exposed to the client along a remote invocation procedure. The API would primarily target single sources for specifics. Some example API would be :-

1. int getNumVoices(String midiFilePath)

2. double melodicContourMovement(String pngSheetFilePath)

3. double compareAverageAttack(String musicXMLFile)

This API can be consumed for specific song analysis. As design this API will not work on a bulk of files like its sql counterpart.

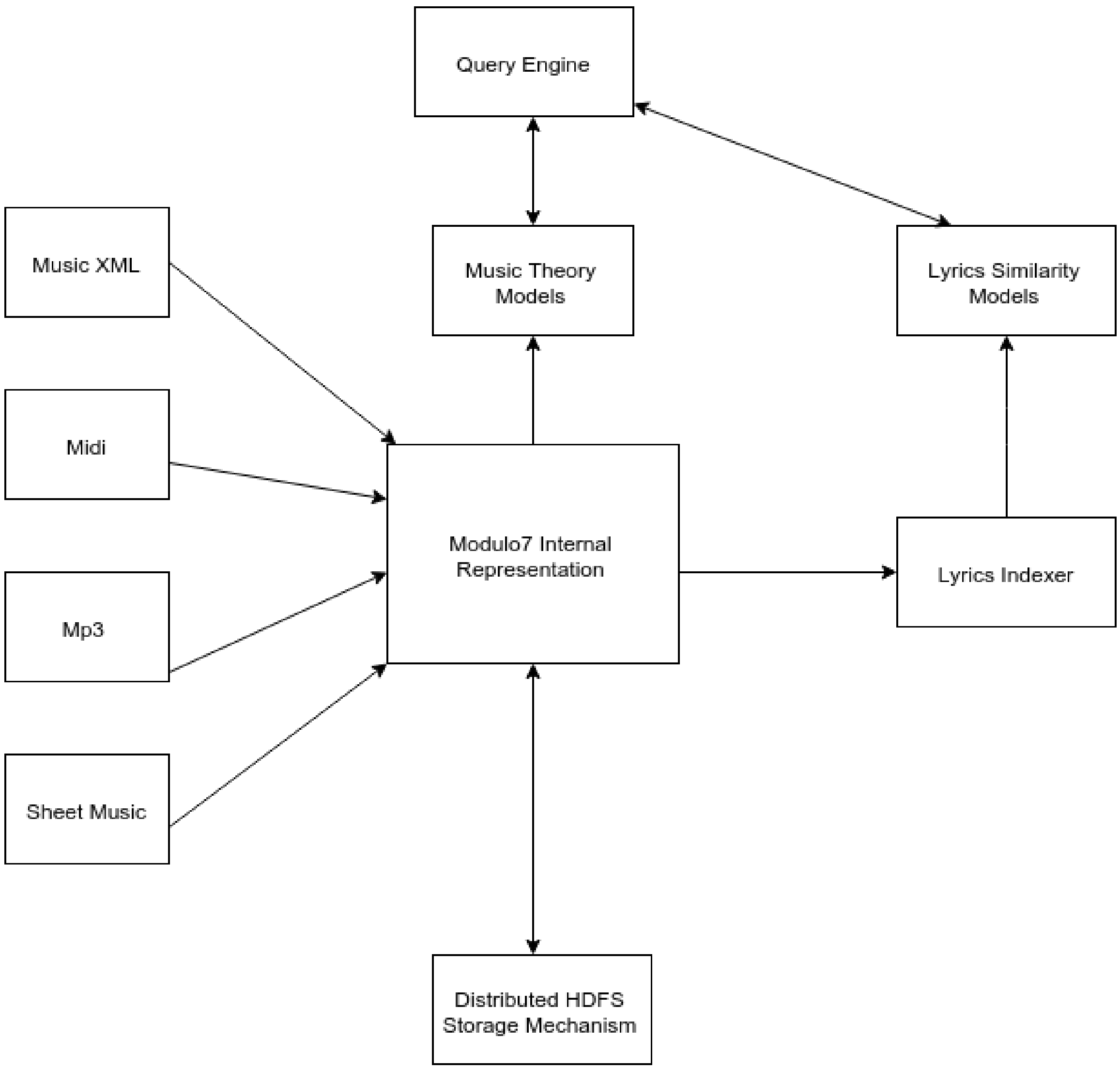

Figure 4.1: Modulo7 architectural design

Chapter 5

Progress Done

The following points of progress have been successfully completed:-

1. Literature study on existing software related to Music Information Retrieval. Software exists which allow feature extraction jMIR.[1] - My primary inspiration, Audio analysis and audio based Information Retrieval - Essentia,[2] marsyas[3], specialized music theoretic exploration of a song Humdrum[4]. None however attack the problem for MIR for scale. Also studied industry's approach on MIR, although companies don't publish all details.

2. In the software architecture part : Completed the domain specific converters for Midi, Mp3 and Music XML along with the Modulo7 internal representation and its binarization. Need to start work on HDFS deployment.

3. Basic Lyrics indexer is implemented(including a tokenizer). Need to start work

on lyrics similarity models.

4. Identified non trivial amounts of datasets to begin testing on (e.g. the million song dataset for specific song features, JHU's Lester Levy Sheet music collection). I have started the process for acquisition of these datasets but working on these datasets would begin on October.

Points to consider and deliberate over:-

1. If certain meta data is not available, estimate that with existing algorithms (e.g if key of a song is not present, should the author include an estimation algorithm for it?).

2. Comparisons with other software. Other software try to address different problems so would have to compare different aspects of each software with Modulo7's components.

3. Criteria for evaluation (speed, accuracy of certain components with existing software etc). Need to figure out what other criteria are appropriate.

4. Investigate alternatively technologies for software design. (Need to be certain this is the best set of tools and design for this problem and this is the best possible architecture).

5. How much breadth coverage is appropriate for the sql algebra space. Is numerical output the only statistics the author should consider or other qualitative

aspects should also be a part of the design space?

6. Should the author attempt special problems already addressed in academia (e.g cover song detection) maybe with a different approach then existing literature?

7. Should the author work on a crawler and how should its scope be defined? While the author has worked on creating crawlers to mine domain specific sources, there might be copyright infringement on mining arbitrary sources. An alternative is to acquire data that is explicitly marked for research.

8. Should the author work on a "imprecise querying system" i.e. a search engine based on music theoretic criteria. This work would be extremely ambitious.

Bibliography

[1] C. McKay, *Automatic Music Classification with jMIR*. Montreal: McGill University, 2010.

[2] *ESSENTIA: an Audio Analysis Library for Music Information Retrieval.*

[3] P. G. Tzanetakis, "Marsyas a framework for audio analysis," *Organized Sound*, vol. 4(3).

[4] *The Humdrum Toolkit: Reference Manual. Menlo Park, California: Center for Computer Assisted Research in the Humanities, 552 pages, ISBN 0-936943-10-6.*