The Chinese University of Hong Kong

Department of Computer Science and Engineering

2017-18 Fall

Final Year Project

Progress Report

KY1701 Automatic Piano Reduction - backend (chord Identification)

Project Progress

In this month, our job focuses on getting familiar with music theories and the python library Music21. We came up with a chord matching algorithm using the naive approach, and bassline analysis which is a possible approach for identifying chord changes.

1. Chord Matching - Naive Approach
   1. Explanations

In the very first stage, we did the chord matching using naive approach. We used two chord tables for the searching - major table and minor tables with all possible chords. Notes of chords are precomputed and stored in each row.

Here is the pseudo code for doing total matching and inclusion matching.

Function TotalMatchAndInclusion(notes)

notes = removeDuplicate(notes)

for chord\_notes in major\_table:

if notes is subset of chord\_notes:

//inclusion

if notes equal chord\_notes:

//totalMatch

//Do same operation for minor\_table

For partial match, we consider cases of one dissonant note and two dissonant notes. As we did not do any melodic analysis algorithm in this stage, we treated every note as having the equal chance of being a dissonant note and tried all possible combinations.

Function oneAndTwoDissonant(notes)

notes\_set = removeOneNote(notes) //return all combination of

notes after removing one

for notes in notes\_set:

notes = removeDuplicate(notes)

for chord\_notes in major\_table:

if notes is subset of chord\_notes:

//inclusion

if notes equal chord\_notes:

//totalMatch

//Do same operation for minor\_table

//Do same operation with removeTwoNotes()

Complexity for searching the chords is O(n) as of linear search.

* 1. Improvements to be done

Time complexity can be further reduced. We are considering using hash table to reduce the time cost in the upcoming future.

In cases of one and two dissonant note, nodes are excluded without considering if they are really the dissonant notes in the melody. We will develop melodic decoration analysis in the coming stage.

1. Bassline

In pieces for musical ensembles, bass line is a significant indicator for chord changes. In this stage, we tried to identify a chord unit using bassline changes. However, the bass instrument is not necessarily playing all the time. We need to search for the lowest playing instrument in every measure.

* 1. Approach

The note range of instruments can help us identify the bass instrument. For every instrument in the score, we use the following function to calculate for the average pitch.

function getAvgPitchOfPart(part)

for note in notes\_in\_part:

totalHeight += note.pitch

return totalHeight/noteCount

We than get a list of part index sorted by the average pitch of the parts.

function getPartList(score)

for part in score.parts:

partList[index] = getAvgPitchOfPart(part)

return sort(partList)

For every measure, loop through the part list to get the bass line. If the lowest instrument does not have any notes on that measure, check for the next one, and so on.

function getBaseline(score)

partList = getPartList(score)

for measure in score.measures:

for part in partList:

if measure\_in\_part have notes:

//lowest part

* 1. Testing Result

Here is one of the testing result of W. A. Mozart, Così fan tutte, KV 588





* 1. Improvements to be done

There are cases that the bass line plays with melodic decorations, i.e. changing the bass note due to changing in chord. (For example, in the bassline of first image).

Pedaling is possible to occur, i.e. having chord change while bass line doesn't change. We will develop another method for dividing chord unit.

Future Development

1. Chord Matching – Hash tree

Using a hash table can reduce the time complexity to O(1). We will develop a multi-dimensional hash tree with each note as the index of each row.

1. Melodic decorations analysis

By analyzing the melody, we can identify the possible non-harmony notes, e.g. passing notes, anticipation, suspension. We will try to develop a scoring system to give different priority to notes when doing Chord Matching.

Bass analysis (e.g. arpeggio, Alberti bass) is another upcoming approach for identifying a chord unit.

1. Chord Unit identification

Other than using the bassline as the indicator, we propose another method for identifying a chord unit. Using a “beat” as the starting chord unit, we will develop an algorithm to decide if division or grouping is needed.

We will update our process and give more detailed explanations in the following Gitbook site:

<https://franceslicy.gitbooks.io/ky1701-chord-identification/content/chord-unit.html>