Support Vector Machine-based Automatic Music Transcription for Transcribing Polyphonic Music into MusicXML

Krisna Fathurahman
Informatics/Computer Science
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
krisna.fathurahman@gmail.com

Dessi Puji Lestari
Informatics/Computer Science
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
dessipuji@stei.itb.ac.id

***Abstract*—Automatic Music Transcription (AMT) which transcribes music into music sheet is a challenging task since it requires combination of three different knowledges: signal processing, machine learning, and musical model. The task is more challenging when AMT applied to the polyphonic music. Such task required the system to recognize the pitch, timbre, tempo, onset, and expression into a readable music sheet. This paper describes our works in building such system. In this research, the most promising and prominent approach is applied. Those are the Mel's Frequency Cepstral Coefficient (MFCC) as the features and the One-against-all Support Vector Machine (SVM) as its decoder. The combination of both methods had shown very promising results. The output of our AMT system is a music sheet in a MusicXML format with high compatibility with music software nowadays.**

Keywords—music transcription, polyphonic music, mel's frequency cepstral coefficient, support vector machine, musicxml

I. Introduction

Music is an art that uses soundwave as its media. Through music, human are able to express their idea or emotion in form of pitchs, chord, rhythms, dynamics, expressions and combinations of all of them. Music can be played by one kind of music instrument (monophonic) or by combining many kinds of music instruments or many band of music instruments (polyphonic). Every music instrument has a particular sound colour or timbre. Timbre can be determinde through pitch and sound intensity that produced by the music instrument itself [1].

In the early 1970, some researchers built systems to trancribe music automatically. However, most of their proposed methods weren't too effective and worked well only for particular domain or instrument. Research on the AMT system has been conducted more than 30 years. However, some problems still become a huge wall of challange especially for polyphonic music. In order to be able to transcribe polyphonic music, an AMT system has to conduct several steps as follows: to determine the pitch and the chord that played in the same time, to count duration of notes, to determine articulation and dynamic of each note, to determine which instrument play the note, to determine the key signature of the music, to determine the tempo, and to represent transcription result into proper and readable format.

MusicXML is a new format for music file exchange and distribution. This format is needed to represent all cases that happened in digital music sheet exchange. MusicXML can record all texts in a music sheet, notes explicitly (different with MIDI that use piano keys model which need to interpret is it E-flat or D-sharp [2]), key signature and time signature changes, tempo changes, tuplet and also more variative percussion notes representation. MusicXML has been used by a lot of music softwares right now. So, a music software or music-based system which could produce a file of MusicXML format is really good to be develop nowadays [2].

II. Related Works

Various approaches have been proposed to build AMT systems for polyphonic music. In 2003, the Petrusa developed a polyphonic music transcription tool by applying the Short-Time Fourier Transform in the front-end for time-frequency analysis. However, the performance of the system is not good due to the linear spectrum that it produced. In 2005, Donovan proposed Wigner Distribution and proven to be powerful method to represent real signals. However Weigner Distribution requires high computational cost. Thus, it is still rare to be applied in a general music analysis system. The Mel Frequency Feature Coefficients (MFCC) still become state-of-the-art for feature extraction method and widely used in speech recognition systems and music analysis systems. MFCC is a multiresolution analysis that is very for extracting feature of soundwave [3].

In the machine learning area, many methods have been applying to classify pitchs or chords of a music. For example, the sequencial probabilistic method such as the Hidden Markov Model (HMM), methods that allow many features as input like the Artificial Neural Network (ANN), and the Support Vector Machine (SVM). For SVM, researchers used various approaches such as the Nonlinear SVM, the One-against-one or One-against-all SVM. There's also good result by using SVM with memory that developed by Constantini et al [4] on 2010 for polyphonic music transcription in piano.

Many music transcription systems were developed until now. Most of those systems only produce predicted label or class such as pitch, chord or duration. There's also music transcription system that produces digital music sheet such as in the MIDI format. However, such format can only represent

978-1-4673-7319-7/15/$31.00 ©2015 IEEE

notes position in piano keys. The AMT system that produce digital music sheets on the MusicXML format is hardly found in music analysis development. Thus, in this research we developed an AMT system that produce produce digital music sheets on the MusicXML format.

III. The Mel's Frequency Cepstral Coefficient (MFCC)

MFCC has become a prominent method for sound processing. MFCC works similar with how human ear work when analyzing the time-frequency of sound signals [5]. There are 7 main steps of taking the MFCC features. Those are: framing, windowing, pre-emphasize, Discrete Fourier Transform (DFT), taking log results, Mel's scaling, and derivation as can be seen in Figure 1.

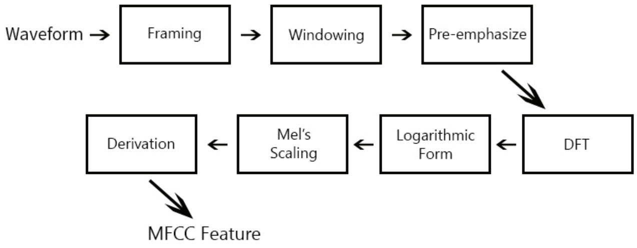

Fig. 1. Mel's Frequency Cepstral Coefficient steps.

Framing is a step to chunk continous soundwave into several discrete frames. Those frames have duration of time around 20-40ms. The frame will have two overlapping section around 5-10ms in the beginning and the end of the frame. Windowing is conducted to group some frames to process signal more efficient. Pre-emphasize is a process to increase the energy level to the signal especially to high frequency signals, so that the high frequency signals can be detected. The most high frequency signals have very low energy level so it is difficult to detect such signals. Then, signal in the time domain will be transformed into frequency domain by applying the DFT to get the soundwave spectrum. The wave spectrums are transformed into logaritmic function to produce inverted wave or "cepstrum". After that, cepstrum will be sorted into Mel's scale, hence we call the result of this step as MFCC feature. For getting complete features, we need the velocity and acceleration information of the soundwave of which we can get it by deriving the MFCC features.

For music analysis, MFCC can be used directly without any modifications. Even though music signals have larger range of frequency than the speech signal, there are no reasons that sound intensity or energy can't be detected logaritmically. Thing to concern is that Mel's scaling may not be good to detect high frequency signals. Beth Logan [5] conducted some experiments to prove that phenomena by comparing the number of segmental errors between the Mel's scaling and the linear spectral model using mixed speech signal and music signal as the test data. As we can see in Table I, the Mel's scaling still has better performance than the linear spectral model.

TABLE I. Experiment result comparing Mel's Scaling and Linear Spectral Model[5]

Mix Number	Spectrum	Segmental Error (%)
4	Mel's Linear	7.1 14.3
8	Mel's Linear	1.8 8.9
16	Mel's Linear	3.6 10.7

IV. The Support Vector Machine (SVM)

SVM is widely used in the sound classification task as well as inn AMT systems. Particularly, it has been successfully used for detecting pitch or detecting onset (position and duration of notes). Practically, SVM has been used in various ways. For example, if the data are linearly separable, then there will be some hyperplanes available that can separate those data into two classes without error. From all those hyperplane, we choose the one that has the most maximum margin, hence we called it the maximum-margin hyperplane or optimum hyperplane (see Figure 2). Such approach becomes the principle of Maximum-Margin SVM (M-M SVM) [3].

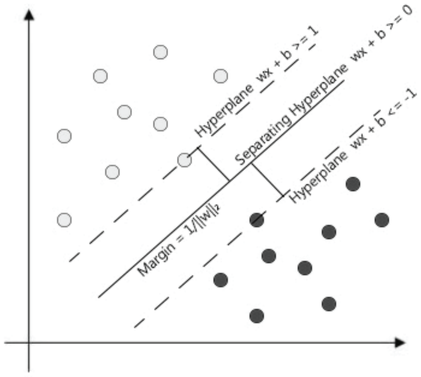

Fig. 2. Maximum Margin Support Vector Machine.

Using the margin size approach, M-M SVM can only handle linearly separable data. In the reality, the training data probably contain some noises. Therefore linear separation solution can not solve this problem. The Nonlinear SVM has been proposed to handle this problem. It could map the non linear data through higher dimension features space to get linear separation. These mapping can be conducted by using the Kernel function. Kernel function was applied to transform input samples into variable product. There are two kinds of Kernel function that mostly used for non-linear mapping: Polynomial and Gaussian Kernel [3]. Those functions are chosen depend on the characteristics of the data.

In the music analysis system, it is possible to have data with superb quantity of classes or more than two classes. For example, a music which plays pitch sound of C-D-E-F-G-A-B has 7 note classes. In such case, the multi-Class SVM is usually applied. The Multi-Class SVM can be applied in two ways: One-against-one and One-against-all. For One-against-all, data with K number of classes will build K number of 2-class pair-wise (a class and its negation) classifier (SVM). Then all classification result from each classifier will pass through decision function by comparing class with margin component of its classifier. In the One-against-one schema, if we have K classes, then there will be (K (K-1))/2 number of pair-wise classifier [3].

V. SYSTEM ARCHITECTURE

Our AMT system consists of three main modules: (1) the feature extraction module using MFCC; (2) the Decoder module that contains One-against-all SVM classifier; and (3) the MusicXML generator module. Figure 5 shown the architecture of our system.

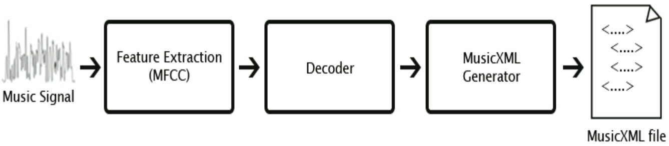

Fig. 3. AMT system architecture design.

Polyphonic music signal in .wav format will be supplied into the system. The features of the music signal are extracted using the MFCC. Then, the decoder will recognize all musical attributes, such as the pitch and cords, tempo, onset, and its duration. After we get the musical attributes, we transform those attributes into music notation using MusicXML generator. Finally, a MusicXML file will be produced by our system. The details explanation of each module will be described below.

A. Feature Extraction

For extracting the features, MFCC was choosen as this method is proved to be very effective in the sound analysis. It could mimic the work of human ears which detects sound frequency exponentially. This module will be build by employing available tool named Hidden Markov Model Toolkit (HTK). Actually HTK is a tool for building an automatic speech recognizer based on the hidden markov models. However, since it has a very useful tool to extract the speech signal feature using MFCC we employed the tools. Here we are only use its MFCC features tool. The input of the system and to this module has to be in a .wav file format and the output of the module is an .mfc file with HTK binary format.

B. Decoder

Inpired by music attributes in music notation, we breakdown the functionality of the decoder into several tasks in order to recognize all music attributes. Therefore, decoder module will consist of several sub-components that become the core function of this module:

1) Pitch and chord detector.
2) Tempo detector.
3) Onset and duration detector.
4) Music instrument detector.
5) Expression and articulation detector.
6) Key signature detector.

There are some sub-components within decoder module that use SVM: pitch and chord detector, instrument detector and expression and articulation detector. Those sub-components will use a tool called LibSVM.

1) *Pitch and chord detector*

We developed the pitch and chord detector employing the SVM that was trained by using the prepared training data. We conduct the One-against-all approach. Feature of frequency and energy will become the attribute of classified data. Output of this component is a pitch label and its octave. For example label "C4" stands for pitch C Major in the fourth octave. In this component, silence will assumed as a rest note and then will be calculated for its duration.

2) *Tempo detector*

The frequency, time, and energy features resulted from the feature extraction of polyphonic music signal are calculated to get an average time of notes being played. This average time become the tempo of the music being played. The tempo will be used as the input in the onset and duration detector to determine the duration of one beat played.

3) *Onset and duration detector*

Label or class resulted from the pitch and chord detector and the tempo resulted from the tempo detector are the inputs of the onset and duration detector. This module calculate and decide what kind of notes is being played by counting repeatitive label divided by duration of a beat note, then produce a label or class of onset and duration. The possible labels are: whole note, half note, fourth note, eighth note or sixteen note.

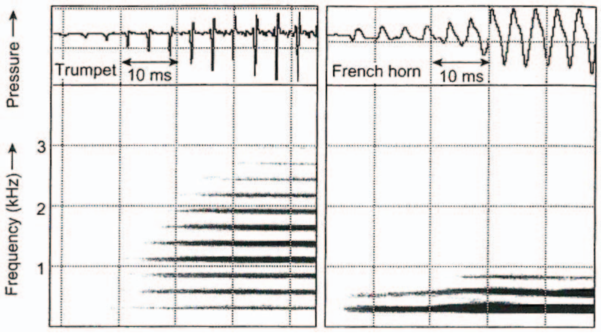

Fig. 4. Trumpet and Franch Horn frequency spectral for C4 note played. [6]

4) *Music instrument detector*

As we can see in Figure 4, for the same pitch (C4), trumpet and french horn has particular frequency spectral according to its instrument. Therefore, to detect music instrument or the timbre, the frequency and the energy feature are used as its parameters. In every window, the timbre of the window will be detected to determine the type of instrument being played.

5) *Expression and articulation detector*

The music expression of a note will be detected by using instruments label and energy feature resulted from the feature extraction. Those music expressions are: f (*forte*), mf (*mezzoforte*), mp (*mezzopiano*), p (*mezzopiano*), ff (*fortesimo*), fff (*fortesisimo*).

6) Key signature detector

Pitch variations that exists in the played music is the input for the key signature detector. Every key signature has their own chromatic pitchs so we can determine what key signature was used. This is conducted to avoid too much additional music attributes such as: the sharp and flat in the resulted music sheet. Using music notation rules, iff more than one key signature are detected, key signature with the highest probability will be chosen.

C. MusicXML Generator

MusicXML generator produce a music sheet in MusicXML format by transforming all musical attributes resulted by all module in the decoder. This musicXML file is generated by following the music notation writing rule. Here, we assume that the time signature being used in music sheet is 4/4. To determine the one measure and the rest, the XML Generator conducts the calculation. In general, this module work by using the rules based on the music notation writing rule and MusicXML grammar. In the end of the process, MusicXML file will be saved in prototype file directory.

VI. Experiments and Analysis

A. Experimental Setup

We collected the data for making the SVM model. Since we need polyphonic music, we need minimum two music players who play some music score together, then to record their performance into a single .wav file. For this experiment we recorded trumpet and baritone to play some score that is being cued by same metronome for maintain their tempo. The use of metronome is important to avoid any human errors that might occur in our experiments.

A complete music play record is then sliced into several .wav music files so that each file contains only one note. We employed a tool called Audacity to slice the file. After those files were collected, we gave the labels for a each file. The labels of each file are the music instruments and the pitch exits in the file. The silence and noise (non-music voice) is also sliced and labeled. In the decoder, this silence and noise data are grouped and trained to build silence model for pitch and chord detector.

B. Experiments Results

Before we conduct the experiment, we gather some music recordings of Trumpet and Baritone play the same music score three times. Tempo used for this music score was 120 BPM using same metronome as reference for each repeatition. 6 .wav files produced and named “set A” to “set F”. Each set was sliced and produced total 50 .wav files of a note or silence in each file. These 50 .wav files became training data.

This system use MFCC as front-end, we configured this method to give us 26 components of feature and 5 ms frame per instance. Each instance was labeled depends on what model that will use it. For example, for pitch detector model, we gave label Bes4 or F4 which means what pitch is that and its octave.

For this system, we developed three kinds of models:

1) silence or noise detector model,
2) music instrument detector model that related to the timbre of music instrument, and
3) pitch detector model.

To evaluate performance of those models, we conducted the 10-fold cross validation using some measurements adapted from information retrieval method: the precision, recall, and f-measure as follows [4]:

$$Recall = \frac{\sum correctly\ transcribed\ notes}{\sum notes\ in\ ground\ truth} \quad (1)$$

$$Precision = \frac{\sum correctly\ transcribed\ notes}{\sum notes\ in\ automatic\ transcription} \quad (2)$$

F-measure = (2 x Recall x Precision) / (Recall + Precision) (3)

In the training process, to get the best configuration, we conducted model optimalization using 20 files chosen randomly from training data (total 50 files) to produce 61 instances, using MFCC feature extraction explained above, as training data. This optimalization process evaluated using 10-fold cross validation by measuring the precision, recall, and f-measure. Additionally, we also count the model accuracy. The results of our experiments can be seen in Table II.

TABLE II. Experiment Result for Model Optimalization

Configurations	Accuracy	Precision	Recall	f-Measure
C-SVM Linear Kernel	88.52%	0.88	0.88	0.88
C-SVM Polynomial Kernel	88.52%	0.88	0.88	0.88
C-SVM Radial Kernel	85.25%	0.73	0.85	0.78
C-SVM Sigmoid Kernel	85.25%	0.73	0.85	0.78
Nu-SVM Linear Kernel	90.16%	0.90	0.90	0.90
Nu-SVM Polynomial Kernel	88.52%	0.88	0.88	0.88
Nu-SVM Radial Kernel	88.52%	0.88	0.88	0.88
Nu-SVM Sigmoid Kernel	88.52%	0.88	0.88	0.88

As we can see in Table II, the best result is achieved by using the Nu-SVM with Linear Kernel function with model accuracy 90.16%, precision 0.90, recall 0.90, and f-measure

0.90. Since the Nu-SVM with Linear Kernel function gave the best result, we use this model configuration in our next experiments to build three kinds of models: the silence detector, the instruments detector, and the pitch detector models. Evaluation result for silence or noise detector model is shown in Table III.

TABLE III. EVALUATION RESULT FOR SILENCE DETECTOR MODEL

Accuration	Precision	Recall	f-Measure
100%	1	1	1

In this experiment, we achieved very good results (100% accuracy) and 1 for f-measure value. The model works very well in detecting the silence part and non-silence part of a speech signals due to very different value of frequency and energy feature of those two conditions.

For the music instrument detector model, the evaluation results are shown in Table IV and for the pitch detector model is shown in Table V.

TABLE IV. EVALUATION RESULT FOR MUSIC INSTRUMENT DETECTOR MODEL

Accuration	Precision	Recall	f-Measure
95.62%	0.96	0.96	0.96

TABLE V. EVALUATION RESULT FOR PITCH DETECTOR MODEL

Accuration	Precision	Recall	f-Measure
93.52%	0.93	0.93	0.93

We use those models to our system and complete our decoder component. Then, a musicXML file can be produced nicely from our system. We use 50 .wav files explained above as training data and use combination of 6 sets sliced recording as input. Because the output are too long so won't be put in this paper.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we have presented the steps to build automatic music transcription system using support vector machine for polyphonic music. We combine mel's frequency cepstral coefficient as the front-end and support vector machine as the decoder to get all musical attributes in the music notation of a music sheet. Our conclusion is that the support vector machine works very well in detecting the silence part of speech signal, recognizing the timbre or the music instruments, and recognizing the pitch.

For the future works, we are going to implement the tempo detector component and the onset detector component. We also will add data to complete our experiment for build the expression and articulation detector model. Finally we also have to implement our MusicXML generator to be able to visualize MusicXML.

ACKNOWLEDGMENT

The authors would like to express our gratitude to School of Electrical Engineering and Informatics Institut Teknologi Bandung for the research fund support. We also would like to appreciate all of the speaker respondents who want to support our experiments.

REFERENCES

[1] Klapuri, A., Müller, M., Richard, G., & Ellis, D.P.W, "Signal Processing for Music Analysis" in *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 6, 2011.

[2] Kuzmich Jr., J, "MusicXML: Preserving Your Music Beyond MIDI", 2012.

[3] Zhou, R, "Feature Extraction of Musical Content for Automatic Music Transcription.", Thesis, Master of Engineering, Chinese Academy of Science, Beijing, China, 2006.

[4] Tavares, T.F., Barbedo, J.G.A., Attux, R., & Lopes, A, "Survey on Automatic Transcription of Music: Historical Overview of Techniques.", Survey Paper, The Brazilian Computer Society, 2013.

[5] Logan, B, "Mel Frequency Cepstral Coefficient for Music Modelling.", Paper, Cambridge Research Laboratory, Cambridge, 2000.

[6] Howard, D. M., Angus, J. A. S, "Acoustics and Psychoacoustics" 4th Edition. Oxford, UK: Focal Press, 2009.