FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF BACHELOR’S THESIS

Title:	Unsupervised Segmentation of Songs in Full Concert Audio
Student:	Petr Nevyhoštěný
Supervisor:	Ing. Tomáš Kalvoda, Ph.D.
Study Programme:	Informatics
Study Branch:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	Until the end of winter semester 2018/19

Instructions

* Get familiar with concepts and techniques in unsupervised audio and music segmentation. Focus on music/non-music segmentation and robustness among various musical genres.
* Propose a strategy based on methods discussed in the survey and described in [1, 2, 3] aiming at the identification of time boundaries of songs in a full-length live concert audio.
* Implement the proposed strategy in the Python programming language using libraries from the SciPy ecosystem and libraries for audio analysis. Publish your implementation as an open-source Python package.
* Evaluate the implementation on a dataset of available full concerts of artists playing various musical genres, measure the segmentation accuracy. Compare the performance of your implementation with other publicly available implementations.

References

[1] THEODOROU, Theodoros, MPORAS, Iosif and FAKOTAKIS, Nikos. An Overview of Automatic Audio Segmentation. International Journal of Information Technology and Computer Science [online]. 2014. Vol. 6, no. 11, p. 1–9. DOI 10.5815/ijitcs.2014.11.01. Available from: http://www.mecs-press.org/ijitcs/ijitcs-v6-n11/v6n11-1.html
[2] SARALA P, ISHWAR V, BELLUR A, MURTHY H. Applause identification and its relevance to archival of Carnatic music. Proceedings of the 2nd CompMusic Workshop [online]. 2012. p. 66-71. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.650.4983
[3] ZHANG Bihong, XIE Lei, YUAN Yougen, MING Huaiping, HUANG Dongyan and SONG Mingli. Deep neural network derived bottleneck features for accurate audio classification. IEEE International Conference on Multimedia & Expo Workshops (ICMEW) [online]. 2016. pp. 1-6. DOI 10.1109/ICMEW.2016.7574769. Available from: http://ieeexplore.ieee.org/document/7574769/

doc. Ing. Jan Janoušek, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague June 10, 2017

Czech Technical University in Prague
Faculty of Information Technology
Department of Theoretical Computer Science

Bachelor's thesis

Unsupervised Segmentation of Songs in Full Concert Audio

Petr Nevyhoštěný

Supervisor: Ing. Tomáš Kalvoda, Ph.D.

14th May 2018

Acknowledgements

I would like to express my sincere gratitude to my supervisor Ing. Tomáš Kalvoda, Ph.D., whose thorough feedback and valuable advice were very helpful during writing this thesis.

Another tremendous thanks belongs to people at the university, where my eyes got open to see many beauties of the computer science field; and to people at Datamole, s. r. o., who are continually proving that work can be fun, and for providing me with computational infrastructure for the experiments.

Many thanks go also to my friends for helping me to have a truly happy life which I enjoy to live. And last but not least, I would like to let my parents know that without their huge support, very little of what I have done and will do would be possible, and I am really grateful to them.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on 14th May 2018 ....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Nevyhoštěný, Petr. *Unsupervised Segmentation of Songs in Full Concert Audio.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2018.

Abstrakt

Desítky milionů celých živých koncertů jsou k dispozici na službách pro sdílení video- a audiozáznamů, a seznamy písní spolu s časovými údaji jsou podstatnou informací, která je s nimi poskytována. Jelikož je ruční anotace repetitivní a časově náročná, automatický nástroj je velmi hodnotný.

Tato bakalářská práce předkládá řešení pro neřízenou segmentaci písní v audiozáznamu koncertu. Podle mých poznatků je to vůbec první pokus o řešení problému takto zadefinovaného. Hranice segmentů jsou určeny pomocí široce používaného testu logaritmického věrohodnostního poměru a tři různé modely pro klasifikaci jsou představeny.

Na shromážděném datasetu, obsahujícím různé hudební žánry a kvality zvuku, dosahuje implementovaný systém 88,92% f-skóre a 81,30% specifičnosti správně označených sekund audio signálu. Celkové výsledky ukazují, že je jeho detekce hranic poměrně úspěšná, a může tedy sloužit jako obstojný základní model k porovnání s budoucími řešeními.

Klíčová slova segmentace písní, celé koncerty, neřízený, transformace s konstantním Q, mel-škálované spektrum, střední kvadratická energie, spektrální centroid, spektrální tok, spektrální plochost, logaritmický věrohodnostní poměr, detekce anomálií

Abstract

Tens of millions of full live concerts are available on video- and audio-sharing services, and set lists with song time annotations are an essential information provided with them. Since the manual annotation is repetitive and time-consuming, an automatic tool is very valuable.

This thesis proposes a solution to the unsupervised song segmentation in full concert audio. To my best knowledge, it is the first attempt to deal with the problem of this particular definition. Segment boundaries are identified by the log-likelihood ratio method and three different models for the classification are introduced.

On the collected dataset, containing various musical genres and audio quality, the implemented system achieves 88.92% f-measure and 81.30% specificity of correctly labeled seconds in audio signal. Overall results show that its boundary detection is relatively successful, and therefore it serves as a decent baseline system for future solutions.

Keywords song segmentation, full concerts, unsupervised, constant-Q transform, mel-scaled spectrum, root-mean-square energy, spectral centroid, spectral flux, spectral flatness, log-likelihood ratio, anomaly detection

Contents

List of Figures

List of Tables

Introduction

In the last few years, tens of millions[1] of full live concerts were uploaded on video-sharing websites like YouTube or Vimeo, and audio-sharing websites like SoundCloud. Video and audio quality of such recordings range from amateur smartphone recordings with a significant amount of noise to professional camera shots with audio taken directly from the mixer.

Viewers and listeners of these full concerts usually request a list of song names with corresponding time boundaries or at least song beginning times. And indeed, these set lists are often attached to them, either put directly into the description by the uploader or contributed by a fan in comments. However, in a lot of cases, especially when the artist is less known or the recording is nonprofessional, these metadata are missing.

An automated identification of set list in such concerts is desirable because the manual labeling is quite time-consuming. Even if you know the artist well, you have to determine the time boundaries. This problem can be divided into two subtasks:

- *Song segmentation* – The goal of this procedure is to split the full-length audio recording into individual songs. The main challenges for a solution are the poor audio quality, where distinction of music and noise can be difficult, and show-related aspects of live performance such as consecutive songs without a break (separation of two consecutive songs is nearly impossible without identification of these songs) or, on the other hand, break in a song for whatever reason.

- *Song identification* – The task here is to identify individual songs. In other words, a solution in some way compares the live version with the database of studio versions of (if possible) all songs of the artist. The key challenges are differences in live and studio performances, for example, in tempo, pitch, sound in general, structure and audio condition.

[1] The search phrases “full concert” and “full set” give about 36 millions and 32 millions, respectively, of results on YouTube at the time of writing this thesis.

It should be noted that this problem is one of the tasks of Music Information Retrieval Evaluation eXchange[2] (MIREX) where many researchers present their results in the music information retrieval (MIR) field. In this thesis, I deal only with the former subtask.

Although the work would be even more beneficial for the end user if the song identification was included, it is still useful since the song segmentation process is a repetitive task which is always time-consuming, even if an annotator knows the artist. It can serve as a cornerstone for another system which deals with the song identification, for example, utilizing already available tools and services (Shazam, MusicID or Gracenote to name a few).

The reason I chose this subtask is mainly that the unsupervised song segmentation is an unexplored area, contrary to the song identification (or music search in general) where plenty of research exists [1]. Furthermore, for functional song identification system, one must have a sufficiently large database of music available, and such database is difficult and resources-heavy to collect in practice.

Problem Definition

In the problem of the unsupervised segmentation of songs in full concert audio, the task is to identify the start and end time boundaries of semantic audio segments, and properly classify these segments as either song or non-song, however, without pretraining a segmentation or classification model. In live performances, time boundaries are often fuzzy, that is, a song can be started and/or ended with an audience entertaining techniques (such as guitar solos, big rock endings, band improvisation continually turning into a song, etc.). It is then difficult to decide whether a music-like audio belongs to a song or not.

In this thesis, I approach this problem in the unsupervised setting, which means that there is no labeled dataset available for training a model. Instead, the task is to segment the audio based merely on the given content itself. The main advantages are that there is no need to collect sufficiently large and diverse dataset intended for training; and that an unsupervised approach – if it is successful – does not suffer from performance degradation on samples not present in training set as might be the case for a supervised approach [2].

On the other hand, there are some considerable challenges for such choice. Without a training dataset, the system must depend solely on assumptions derived from the prior knowledge of signal and spectral characteristics of music, brought by the creator of the model. Also, the “training” size might be insufficient for feature learning approaches. Because the solution operates only on the audio of given concert, the amount of information available for training an automatic feature extractor might not be enough to catch complex

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

characteristics of the input and therefore to converge to a good and robust representation resulting in useful features.

Motivation

My motivation is to create a tool which identifies the song boundaries in full concert audio and is easy to use for the end users. I make the source code publicly available so it can be extended in the future, either by myself or by members of the community. I believe that it has potential to attract users' and developers' attention, because the multimedia content is a large part of our lives, and song list of full concerts is among the interests of a large number of people. Due to the ability of multimedia consumers to share their own content (thanks to multimedia sharing websites), there is a growing need for automatic processing and analysis tools.

I feel an obligation to justify the decision that my work is taken as an unsupervised problem. My goal is to make my tool to be able to handle the wide range of musical genres, and possibly audio of various quality, since it is mainly intended to function together with user-shared concerts on websites like YouTube or Vimeo. I suppose that training some model in a supervised way would require a significantly huge amount of training data, and collecting it is labor-intensive and time-consuming, and requires cautious treatment due to the copyright laws.

Related Work

In this section, I try to summarize work which has been done in the field of MIR and is more or less related to my work in this thesis. The following paragraphs are only a brief overview and listing of methods without any deep analysis. Some concepts and techniques mentioned here are described and examined in Chapter 1.

General audio segmentation has many applications, for example, separation of sections (speech, music, movie) in radio [3, 4] and TV broadcasts [5], speaker change detection [6], or music structure segmentation [7, 8], among others. Therefore, various techniques have been developed.

Two essential phases of the audio signal segmentation are feature extraction from audio and identification of segment boundaries. Among the most popular features for these tasks are *mel-frequency cepstral coefficients* (MFCC), tonality coefficients such as *linear prediction coefficients*, *zero crossing rate* (ZCR) of the signal, and various spectral features such as *energy*, *spectral centroid*, *spectral flatness* or *spectral flux* [9, 10]. Recently, feature learning approaches using *deep learning* techniques [11] or *spherical K-means* [12] achieved success in various MIR tasks, because they overcome some weaknesses of hand-crafted features, such as the scalability (the possibility to use it in any task) or the limitation to short-time analysis, which is common for such approaches [2].

There are two main categories for the identification of segment boundaries. One approach is to measure a distance between successive windows and find peaks in resulting function of time. The *Bayesian information criterion* (BIC) is found in the literature probably the most often, followed by its simplifications like *generalized likelihood ratio* (GLR) or *log-likelihood ratio* (LLR), and *Kullback-Leibler divergence.* [9]

The second technique is based on machine learning models which are trained to classify each frame of audio. Two most common algorithms are *Gaussian mixture model* (GMM) and *hidden Markov model* (HMM) hybrid, and *support vector machine* (SVM) [9]. As with the feature learning, neural networks have been getting attention in the last few years thanks to their significant successes [13].

According to [14], relatively little work has been done on handling the audio content of full concerts. In the work, they present a solution to the problem of set list identification (i.e., song segmentation and song identification together). Their greedy algorithm takes an excerpt from the beginning of unprocessed audio and compares it with each song in the artist's database. From few top candidates, it then selects the best one and estimates the boundaries based on the studio version of the song. Because the solution is based on a database of studio songs and the segmentation is approached trivially, it cannot be an inspiration for my work. Authors also presented this problem as a new task in MIREX, however, there are no participants other than the authors yet.

Applause identification is being solved in [10], although described techniques are applied solely on Carnatic music (a subgenre of Indian classical music), and not tried on western music concerts. Authors observe quite an intuitive fact, that spectra are more flat for applauses than for music. This knowledge is generally projected into the features like spectral entropy or spectral flatness.

Approaches for speech/non-speech segmentation in user-produced videos are presented in [13]. According to authors, little research has been done on consumer-produced audio. This type of content brings considerable difficulties in processing, for instance, variance in audio quality, background noise or specific content defects caused by the equipment or events happened to the author during the recording. However, user-produced content is significantly growing and already a huge part of the media (as shown in numbers at the very beginning of this thesis) and therefore deserves the appropriate attention of researchers.

I was searching for a literature where authors solve the task as I defined in the Problem Definition. To my knowledge, there does not exist such a work, and therefore my thesis is the first contribution to the problem in this exact definition, with similar problems being solved in the literature listed in the paragraphs above.

Contributions

The goal of this thesis was to develop a system which deals with the song segmentation in full concert audio without any pretraining or knowledge about the content of the concert. As far as I know, it is the first attempt to solve such defined problem. The proposed method draws inspiration from solutions of similar tasks and combines them with the knowledge gained during experimentation and analysis.

Four spectral features are utilized for the segmentation and classification, namely root-mean-square energy, spectral centroid, spectral flux and spectral flatness. Although commonly used in supervised classification tasks, their use in class discrimination without pretraining a model is not so common, since unsupervised audio classification in general is quite an unusual task.

Incorporating the prior knowledge about the features' properties, three models are presented. Silence detection is widely used in speech processing tasks, but its adoption in music-related problems is limited by the scope of the applicability. Detection of parts between songs in concerts is one of a few such applications. Next, a simple threshold-based technique utilizing all four features and their behavior in audio classes of interest is proposed. The third model employs the nearest neighbor anomaly detection technique on the space of features. Since unsupervised song segmentation is an unexplored problem, all these models are applied to it probably for the first time.

All the code implemented in this thesis is open-sourced[3] and distributed as a Python package. Despite its only partial success rate, it can be used for practical purposes. Furthermore, as probably first attempt on this problem, it can serve as a perfect baseline for a future research, especially because techniques employed in this thesis are straightforward and not very complex, or are widely used in audio signal processing.

To evaluate the performance of the presented system, a dataset of full concerts and their time annotations was collected and published as a Python package, which also provides the performance measurement utilities. It is separated from the source code of this thesis in order to offer an independent evaluation framework for the unsupervised song segmentation in full concert audio. A public dataset is also important for results reproduction and quantitative comparison of various works.

Organization

The rest of this thesis is organized as follows:

[3] `https://github.com/pnevyk/segson`

- Chapter 1 presents the architecture of the system and describes the underlying concepts and topics. It starts with an elementary characterization of the sound and its properties and continues with a discussion about the characteristics of music. Then, it presents techniques of representing the audio signal in the time-frequency domain and describes following stages of the pipeline, namely feature extraction, normalization, segmentation, classification, and postprocessing.

- Chapter 2 presents the dataset used in evaluation and metrics employed for measuring the performance. Then, it describes the implementation and discusses the practical applicability of the system. The results and the influence of different parameters are demonstrated at the end of the chapter.

- Chapter 3 discusses the results, provides explanations and highlights the important findings. Some recommendations for future research are given afterward.

Chapter 1

Theoretical Background

This chapter quite thoroughly defines and describe the concepts and techniques which my system is based on. Since my work is rather an application of available methods, and not a research in a specific area, the description of the following topics is not as deep as that which could be found in the literature focused primarily on the terms themselves. However, it should still give a strong intuition of the internals of my work.

The architecture of the system's pipeline is based on those commonly found in the literature, indeed, it is the most straightforward and intuitive way how to design it for this kind of problems. Diagram of the architecture is shown in Figure 1.1.

1.1 Audio Signal Processing

The following subsections draw information from [15, 16, 17, 18, 19], which more deeply describe concepts outlined here. I recommend going through them for the more detailed explanation.

1.1.1 Audio Signal and Its Properties

What humans perceive as sound corresponds to air pressure variations near the eardrums. Larger variations cause louder sounds, and faster variations cause sounds higher in pitch. The air pressure varies with time continuously, and it has a precise measurable value at any point in the time. Therefore, the sound can be represented as a mathematical function and is often referred to as a *continuous signal.*

As already noted, the size of the variations corresponds to the loudness of the sound. If the air pressure is constant or the size of the variations is below a frequency-dependent threshold, no sound is being heard. With increasing magnitude, the sound is louder, and even short exposure to variations of size above some certain threshold can lead to the hearing damage.

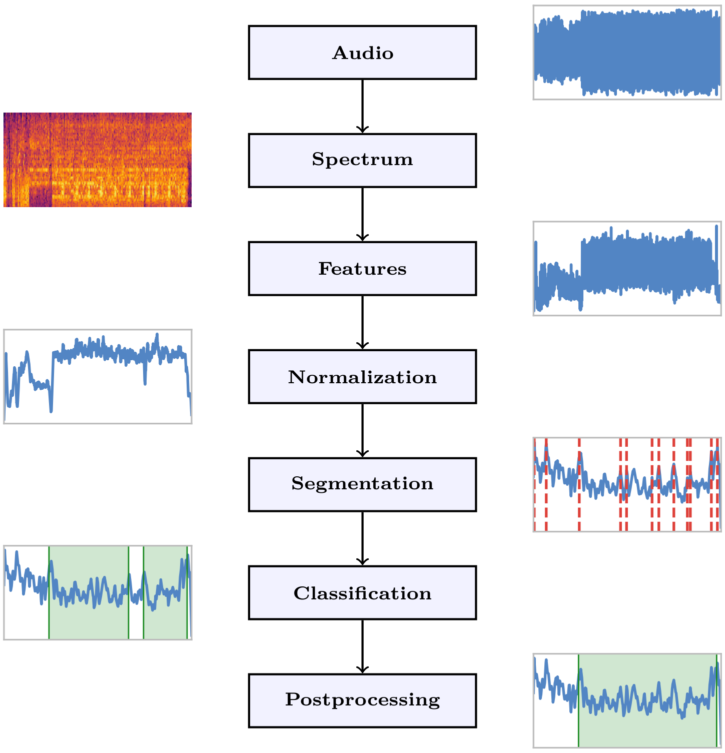


Figure 1.1: Architecture of the system.

Air pressure is measured by the SI derived unit Pascal (Pa) which corresponds to N/m^2 (force/area). The usual value at the sea level is 101 325 Pa. The term *sound pressure* then represents pressure variations relative to the surrounding air pressure. Because human ears – given that attribute by evolution [20] – perceive sound logarithmically, it is common to express the sound pressure at the logarithmic scale, in the unit of decibels (dB), which is defined by

$$L_p = 10 \log_{10}\left(\frac{p^2}{p_0^2}\right) = 20 \log_{10}\left(\frac{p}{p_0}\right) , \tag{1.1}$$

where p is the measured sound pressure and p_0 is the reference sound pressure, usually equal to 0.00002 Pa as this value corresponds to the quietest sound audible by humans. Square of the sound pressure (used in the equation) represents the *power* of the sound which relates to the loudness as perceived by humans. Note that decibel is the dimensionless quantity and has no physical unit because it is the ratio of two pressures. Table 1.1 shows few sound sources and important levels of the sound pressure in Pascals and decibels.

The second fundamental property of the sound is its *pitch* which corresponds to the frequency (speed) of the variations of the air pressure. If $f(t)$ is

	Pa	**dB**
Jet aircraft (50 m distance)	200	140
Threshold of pain	63.2	130
Threshold of discomfort	20	120
Disco (1 m distance)	2	100
Conversation (1 m distance)	0.02	60
Rustling leaves in the distance	0.000063	10
Threshold of hearing	0.00002	0

Table 1.1: Examples of sound sources and important sensation thresholds with corresponding sound pressure in Pascals (Pa) and decibels (dB). Note that the distance from the sound source is important, because the air pressure variations decrease with the distance.

a periodic function of time t with the period $T = \frac{1}{\nu}$ (i.e., $f(t+T) = f(t)$ holds for every t and T is the smallest positive number with such a property), then we say that $f(t)$ has the frequency ν. Frequency is usually measured in Hertz (Hz) where 1 Hz is equal to one cycle per second. The human ear is able to perceive sounds with a frequency between 20 Hz and 20 kHz[4].

Audio waveforms can be either periodic or aperiodic. Periodic waveforms are complex tones consisting of a fundamental frequency and a series of overtones (frequencies higher than the fundamental) and/or multiples of the fundamental frequency. These sounds can be played by the vast majority of musical instruments (e.g., string and wind instruments) and are perceived by the listener as clearly defined pitch or their combination.

On the other hand, aperiodic waveforms are created by non-harmonically related sine tones (more deeply discussed in the Subsection 1.1.2). Percussion instruments are an example of the source of such sounds. Both tonal and noise-like properties can contribute to a sound. For example, speech consists of voiced, tone-like signals (vowels such as /a/ or /i/) and unvoiced, noise-like signals (consonants such as /f/ or /s/) [21], whereas music is usually composed of notes and therefore is highly tonal.

Although the audio signal is a physical phenomenon and therefore is continuous by nature, we usually need to represent it in digital devices, which are on the contrary discrete. The *discrete audio signal* is a sequence $\mathbf{x} = (x_n)_{n=0}^{N-1}$ whose values correspond to measurements of the air pressure of a sound, recorded at a fixed frequency (called *sampling rate*) and mapped into a countable set of numbers (this process is called *quantization*). In the real world, a widely used sampling rate is 44 100 Hz with 16-bit number format. However, it can be observed that the most information is present in lower frequencies, and hence a significantly smaller sampling rate can be used for automatic audio analysis resulting in a faster computation where unnecessary details are missing.

[4] The range is getting narrower with the age.

1.1.2 Music, Melody, Harmony and Rhythm

An exact definition of music is problematic, but there are usually some properties which the majority of people recognize as the properties of music. As the main goal of music is, arguably, to please its listeners, it should reflect specific aspects of human auditory perception.

One fact is that two signals whose frequencies fall into ratio 2:1 (which is called *octave*) are perceived as highly similar. There are other ratios which share many harmonics and are therefore perceived as similar and pleasant to listen. These are called consonant harmonies. Examples include 3:2 or 4:3 (in general, ratios where the nominator and denominator is a small integer).

The sequence of notes at a certain pitch with a certain duration forms the *melody*. Melody is one of the essential features of music recognized by the listeners. Another one is *harmony*, which is a combination of simultaneously played notes at different pitches, usually in consonant ratios. One important note is that the ubiquity of simultaneous pitches sharing harmonics is a major challenge in automatic music analysis.

The last fundamental property of music – somewhat orthogonal to melody and harmony – is *rhythm*. It is described by *beat* which intuitively corresponds to a sequence of significant pulses (note onsets or percussive events) that are regularly spaced in time. The rate of these pulses then refers to the term *tempo*. Both terms contribute to the rhythmic dimension of music and therefore are very important for the listeners.

1.1.3 Fourier Transform

This subsection briefly summarizes the mathematical foundations of *Fourier analysis* for the purposes of this thesis. I refer to [18] or [15] for more thorough description and formal proofs. Fourier's theorem says that any reasonably well-behaved function – while the meaning of "reasonably well-behaved" is out of the scope of this thesis – defined on some interval can be decomposed into a sum of sines and cosines, or equivalently, a sum of complex exponential functions.

In audio signal processing, the discrete variant of Fourier transform is used since the processing is performed on discrete signals. Recall that discrete signal is a vector $\mathbf{x} = (x_n)_{n=0}^{N-1}$, its *discrete Fourier transform* (DFT) is a vector $\tilde{\mathbf{x}}$ defined as

$$\tilde{x}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi ikn}{N}}, \tag{1.2}$$

where $\tilde{x}_k$ are called DFT (complex) coefficients, and $k = 0, 1, \ldots, N-1$.

The *absolute value* of $\tilde{x}_k$ corresponds to the amount of contribution of the complex sinusoid at the frequency $k\frac{f_{\text{sr}}}{N}$, where f_{sr} is the sampling rate. The spectrum of absolute values is called *magnitude spectrum* (or sometimes amplitude spectrum), and its second power is called *power spectrum*[5]. It can be seen that both sampling rate and length of the signal affect the frequency resolution of discrete Fourier transform.

The *argument* of $\tilde{x}_k$ corresponds to the phase – or in other words, the shift or delay – of the sinusoid. Note that in signal processing, the factor $\frac{1}{\sqrt{N}}$ is usually omitted because it is useful only in the mathematical point of view and presents an unnecessary computational overhead.

One important property of discrete Fourier transform is that if $\mathbf{x}$ is a real-valued vector (which is the case in signal processing), the following equality applies:

$$\tilde{x}_{N-n} = \overline{\tilde{x}_n} \quad \text{for } 1 \leq n \leq N-1\,, \tag{1.3}$$

where $\overline{z}$ denotes the complex conjugate of a complex number z. That is, the half of the coefficients is redundant and is usually omitted in practical applications.

It should be noted that with the naive implementation by definition, the computational complexity of DFT is $\mathcal{O}(N^2)$, which is unfeasible for larger, or many, inputs. However, an algorithm for its fast computation, called *fast Fourier transform* (FFT), was developed and with it, one is able to compute DFT in $\mathcal{O}(N \log N)$. Therefore it is usable in practical applications.

1.1.4 Time-Frequency Representations

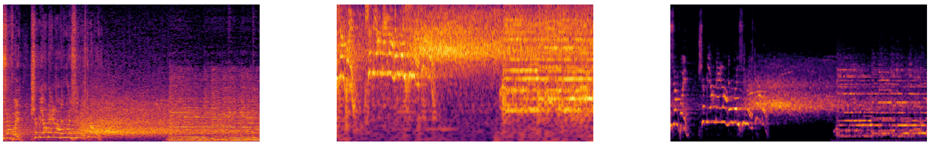
Two essential properties of the sound are the spectral composition (frequency content) and the temporal dimension. Both these properties are highly relevant to the perception of sound and therefore to audio analysis, so it is natural to describe the audio signal in both terms jointly.

Spectrum is usually obtained by Fourier transform of the signal which produces a complex-valued vector. Each value of the vector corresponds to one frequency band whose properties depends on the transformation settings. The phase of the short-time spectrum is considered insignificant for practical applications and therefore is often omitted.

Although the audio signal is nonstationary (its properties change over time), automatic analysis usually assumes that the signal properties change relatively slowly with time. Following this assumption, spectral properties of the signal are computed for short – usually overlapping – windows. Concatenation of these frequency representations of windows (referred to as *frames*) then forms a time-frequency representation of the signal called *spectrogram*. Examples can be seen in Figure 1.2.

Determining the window length represents the trade-off between the frequency resolution of relatively stable content and the time resolution of rapid changes. The length in time units depends on the sampling rate and the win-

[5] The term relates to power of the sound as mentioned in Subsection 1.1.1.



(a) STFT **(b)** CQT **(c)** Mel-scaled spectrum

Figure 1.2: Various spectrograms of the same excerpt of audio signal which contains the start of a song. Horizontal axes correspond to the time and vertical to frequency bins of a spectrum.

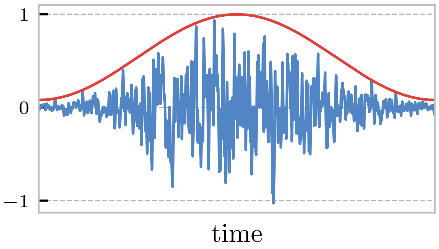


Figure 1.3: Visualization of Hamming window function (red line).

dow length in the number of samples and is typically in tens of milliseconds[6]. Also, windows are usually overlapped by 50% or more.

It is advisable to multiply the signal of the window by a *window function* before applying the Fourier transform. The reason is that the process of windowing creates so-called spectral leakage, in other words, the emergence of artificial frequencies not present in the original input. Among others, Hamming window is an example of such a window function, defined as $w(n) = 0.54 - 0.46\cos\frac{2\pi n}{N_w}$, where N_w denotes the length of the window. Note that these functions have usually very similar shape (visualization of Hamming window is in Figure 1.3) and differ only in details.

Short-Time Fourier Transform

Short-time Fourier transform (STFT) is the most popular spectrum computation method and directly derives from the Fourier analysis. It is just discrete Fourier transform computed on a short window.

Let $\mathbf{x} = (x_n)_{n=0}^{N-1}$ be a discrete audio signal represented as a vector and let $\mathbf{w} = (w_j)_{j=0}^{N_w-1}$ be a vector corresponding to a window function of choice, then

[6] For example, when the sampling rate f_{sr} equals to 44 100 Hz and the window length in the number of samples N_w is 4096, then the window length equals to $\frac{f_{\text{sr}}}{N_w} \approx 92.9$ ms.

STFT is given by

$$X^{\text{STFT}}(k,n) = \sum_{j=0}^{N_w-1} \mathbf{w}_j \mathbf{x}_{n+j} e^{\frac{-2\pi ikj}{N_w}}\,, \tag{1.4}$$

where $k = 0, 1, \ldots, \frac{N_w}{2}$ and $n = 0, 1, \ldots, N - N_w - 1$, and N_w is chosen to be an even number. The reason why k goes only to $\frac{N_w}{2}$ lays in Equation (1.3).

Notice that the definition directly resembles Equation (1.2). $X(k,n)$ corresponds to the window beginning at $\frac{n}{f_{\text{sr}}}$ in seconds, and frequency $k\frac{f_{\text{sr}}}{N_w}$ in Hertz, where the bandwidth of k-th bin is equal to $\Delta_k^{\text{STFT}} = \frac{f_{\text{sr}}}{N_w}$.

Constant-Q Transform

Frequency bins of STFT are linearly distributed over the spectrum, that said, each bin has exactly the same width as others. However, as already mentioned, human perception of sound is rather related to the logarithmic frequency scale, and commonly used musical scales follow this fact. *Constant-Q transform* (CQT) transforms the signal from the time domain into the frequency domain so that the center frequencies of the bins are geometrically spaced.

The parameters of the constant-Q transform are also being closer to musical theory, in particular, there are parameters b denoting the number of frequency bins per octave and $f_{\min}, f_{\max}$ specifying frequencies of the lowest and the highest bin, respectively. These boundaries are usually set to represent particular musical notes.

The name of the transform originates in so-called Q-factor – the ratio of the center frequencies to bandwidths. In this case, the factor is equal for all bins, that is, constant. Since bandwidth in constant-Q transform is dependent on frequency k, there is a need to adjust the window length for each bin to reach this property.

The center frequency of k-th bin is given by

$$f_k = f_{\min} \cdot 2^{\frac{k}{b}}, \quad \text{where } k = 0, 1, \ldots, \left\lceil b \log_2 \left(\frac{f_{\max}}{f_{\min}} \right) \right\rceil\,. \tag{1.5}$$

The Q-factor is computed as follows:

$$Q = \frac{f_k}{\Delta_k^{\text{CQT}}} = \frac{f_k}{f_{k+1} - f_k} = \left(2^{\frac{1}{b}} - 1\right)^{-1}\,. \tag{1.6}$$

The last needed component for the constant-Q transform is the window length used to compute bin f_k:

$$N_k = \left\lceil \frac{f_{\text{sr}}}{\Delta_k^{\text{CQT}}} \right\rceil = \left\lceil Q \frac{f_{\text{sr}}}{f_k} \right\rceil\,. \tag{1.7}$$

Let $\mathbf{w}^{N_k} = (w_j)_{j=0}^{N_k-1}$ be a vector corresponding to a window function of choice. Constant-Q transform is then defined by

$$X^{\text{CQT}}(k,n) = \frac{1}{N_k} \sum_{j=0}^{N_k-1} \mathbf{w}_j^{N_k} \mathbf{x}_{n+j} e^{\frac{-2\pi i Q j}{N_k}} \,. \tag{1.8}$$

It should be noted that the term $\frac{1}{N_k}$ serves as a normalization factor. It is present in the Fourier transform literature as well (i.e., it could be present in Equation (1.4)), but is often omitted in practice because it brings unnecessary computational overhead. However, in the constant-Q transform, it plays a little bit different role since the window varies in length for different frequency bins, and is mandatory for the transform to behave correctly.

Mel-Scaled Spectrum

Mel-scaled spectrum is created by mapping STFT spectrum to the *mel scale*[7]. This scale arose from listening experiments in human interpretation of a pitch in the 1930s. It has been observed that the human auditory system perceives the pitch as linear in the frequency range 0–1000 Hz, and as logarithmic when the frequency is higher than 1000 Hz. Later, the change point has been adjusted and various formulas have been introduced. Probably the most popular is

$$M(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \,, \tag{1.9}$$

with the inverse

$$M^{-1}(m) = 700 \cdot \left(10^{\frac{m}{2595}} - 1\right) \,. \tag{1.10}$$

For the purpose of mapping frequencies to the mel scale, mel filterbank is created and then used to effectively do the scaling operation. The parameters of the mel-scaled spectrum, apart from the input spectrum, are f_{min} and f_{max} representing the frequency range, and the number of mel bins L[8].

Let m_l be the function from mel bin index l to the space of mels, bounded by minimum and maximum frequency, defined by

$$m_l = M(f_{\text{min}}) + l \cdot \frac{M(f_{\text{max}}) - M(f_{\text{min}})}{L+1} \,. \tag{1.11}$$

Two artificial mel bin centers, which represent the range boundaries, must be created, and for that reason, the variable l goes from 0 to $L+1$. Because the frequency resolution in mel space is usually lower than the resolution of STFT, there is the need to round mel frequencies into the nearest STFT bin.

[7] The name “mel” originates in the word “melody”, indicating that it relates to the sound pitch as human perceives it.

[8] The number of mel bins is usually chosen to be in the range 20–50, sometimes exceeding this interval.

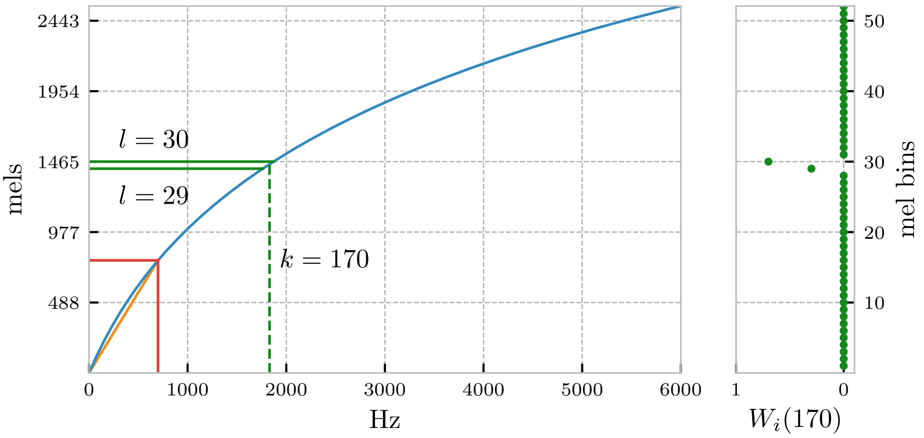


Figure 1.4: Blue line corresponds to the mel scale mapping from Hertz to mels. Red lines denote the area where the mel scale is supposed to behave nearly linearly (illustrated by orange line). For k-th STFT bin (green dashed line) there are shown the closest neighbors on the mel axis (solid green lines). On the right, there is the function $W_i(k)$ for that particular k from which it is visible how much i-th mel bin contributes to the k-th frequency bin.

The following function is used to compute frequency bin index from mel bin index:

$$f_l = \left\lfloor \frac{N_w M^{-1}(m_l)}{f_{\text{sr}}} \right\rfloor . \tag{1.12}$$

The last step is defining a collection of weight functions (represented as a matrix in practice) which properly weigh the contribution of mel bin i to STFT bin k:

$$W_i(k) = \begin{cases} 0 & k < f_{i-1} \\ \dfrac{k - f_{i-1}}{f_i - f_{i-1}} & f_{i-1} \leq k \leq f_i \\ \dfrac{f_{i+1} - k}{f_{i+1} - f_i} & f_i < k \leq f_{i+1} \\ 0 & k > f_{i+1} \end{cases} \tag{1.13}$$

where $i = 1, 2, \dots, L$ and $k = 0, 1, \dots, \frac{N_w}{2}$. The intuition behind these weight functions is fairly simple. For two mel bins nearest to given k-th frequency bin, the weight of the contribution of these mel bins is computed proportionally to their distance (the second and the third case in Equation (1.13)). Notice that in the special case when $k = f_i$, corresponding weight is equal to 1. For all other mel bins other than the closest neighbors, the weight is set to zero. Figure 1.4 shows this intuition in a graphical interpretation.

With the mel filterbank $W_i(k)$, it is then possible to construct the mel-scaled spectrum by mapping the power spectrum (i.e., squared absolute values of the complex-valued STFT spectrum) to mel frequencies as follows:

$$X^{\mathrm{Mel}}(k,n) = \sum_{i=1}^{L} W_i(k)\,|X^{\mathrm{STFT}}(k,n)|^2\,. \tag{1.14}$$

Mel scale is also one of the cornerstones of mel-frequency cepstral coefficients (MFCC) which are heavily used in speech analysis tasks (mainly in speech recognition), and lately also in music analysis problems. However, they are primarily used to encode timbre ("sound color") and discards the pitch information [12]. Therefore I cannot use them as features directly, because it is not straightforward how they behave in a song and outside a song.

1.2 Feature Extraction

In this section, I introduce several feature extraction techniques found in the literature, whose properties are promising – and in the experiments more or less proven – to be helpful in song/non-song discrimination. While spectral and time representation of audio signal could be possibly used for the classification directly, they have high dimensionality and low discriminative power [16], and therefore they would exhibit a poor performance.

A popular approach for feature design is utilizing the prior knowledge about audio signals and their spectral characteristics for computing a proper low-dimensional representation which is able to characterize the signal for a given task. A set of features is well-designed when an audio class – song/non-song in my case – can be determined from its values.

In the rest of this section, $X(k,n)$ denotes any of the spectrum functions defined in Subsection 1.1.4.

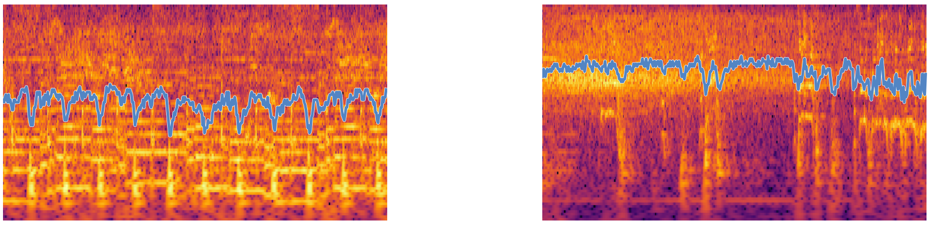
Root-Mean-Square Energy

Root-mean-square energy (RMS) is a simple characteristic which measures the energy in a signal, therefore corresponds to its loudness. As loudness is the most basic and very common discriminator of songs and parts between songs, this feature is suitable for the task.

RMS value can be computed either from the time domain directly or from a computed spectrum:

$$\mathrm{RMS}(n) = \sqrt{\frac{1}{N_w}\sum_{j=0}^{N_w-1} \mathbf{x}_{n+j}^2} \tag{1.15}$$

$$= \sqrt{\frac{1}{K}\sum_{k=0}^{K-1} |X(k,n)|^2}\,, \tag{1.16}$$



(a) Music

(b) Applause

Figure 1.5: Two examples of CQT spectra and corresponding spectral centroid.

where N_w is the window length in the number of samples, $\mathbf{x}$ represents the discrete audio signal, and K is the number of frequency bins of a spectrum $X(k, n)$.

Note that this equality holds true only in the case of STFT spectrum and with no window function applied during its computation. In practice, Equation (1.15) is used if no spectrum is used in the task (which is not usually the case), whereas Equation (1.16) is used when a spectrum is computed for further analysis. The reason is that computation from a spectrum yields more accurate representation due to advantages of the window function and allows to use a different spectrum other than STFT.

Spectral Centroid

Spectral centroid represents the center of the magnitude spectrum, that said, it is the weighted arithmetic mean of magnitudes in frequency bins, given by

$$\text{Centroid}(n) = \frac{\sum_{k=0}^{K-1} f_k \, |X(k,n)|}{\sum_{k=0}^{K-1} |X(k,n)|} \,, \tag{1.17}$$

where f_k denotes the frequency of k-th bin. Greater high-frequency content causes higher spectral centroid, which is illustrated in Figure 1.5. While the major concentration of energy in music lays in lower frequencies, applause and similar noise-like sounds exhibit higher centroid.

Note that this feature somehow correlates with widely used zero crossing rate, which measures the number of zero-crossings in the audio signal in the time domain. ZCR can be called as a measure of dominant frequency [22] in the audio signal, and hence is very closely related to the spectral centroid.

Spectral Flux

Spectral flux characterizes the rate of change between two consecutive frames, or in other words, how quickly the power spectrum changes. It is defined as

$$\mathrm{Flux}(n) = \sum_{k=0}^{K-1} \big(|X(k,n)| - |X(k,n-1)| \big)^2 \,, \tag{1.18}$$

where $Flux(0) := 0$. A high value of spectral flux indicates a significant change in the shape of spectral magnitudes. As music is more structured, it usually features higher amount of change in the bins, and therefore its value of spectral flux is usually higher.

In [10], authors presented a variant of spectral flux, where each spectral frame is divided by a maximum magnitude value of the frame. All values of a spectrum are therefore scaled into the range $[0, 1]$, so only the relative contribution of each frequency bin is preserved. The purpose of this normalization is that if a spectrum is flat, then all values are close to 1 and the change between adjacent frames is small. On the other hand, changes between structured frames result in a high value, which should be the case for music. In this variant, called *peak-normalized spectral flux*, instead of using a spectrum $X(k,n)$ directly, it is first normalized as follows:

$$\hat{X}(k,n) = \frac{X(k,n)}{\max_{k'} X(k',n)} \,. \tag{1.19}$$

Spectral Flatness

Spectral flatness is used to estimate how much is a sound signal noise-like or tone-like, and is defined as the ratio of the geometric mean and the arithmetic mean of the power spectrum [23]:

$$\mathrm{Flatness}(n) = \frac{\exp\left(\frac{1}{K} \sum_{k=0}^{K-1} \ln |X(k,n)|^2 \right)}{\frac{1}{K} \sum_{k=0}^{K-1} |X(k,n)|^2} \,. \tag{1.20}$$

Instead of using traditional formula, the exponential of the arithmetic mean of logarithms is used for the geometric mean in the equation. The equality

$$\sqrt[K]{\left(\prod_{k=0}^{K-1} |X(k,n)|^2 \right)} = \exp\left(\frac{1}{K} \sum_{k=0}^{K-1} \ln |X(k,n)|^2 \right) \tag{1.21}$$

derives from the application of algebraic identities, and is completely legitimate here, because the value of the power spectrum cannot be negative (since it is a square). In practice, values below some “noise” threshold (e.g., $1 \cdot 10^{-10}$) are replaced with that threshold value to handle zeros in logarithm and to avoid floating point number errors.

Value of spectral flatness is always from the range $(0, 1)$, while values close to zero correspond to structured, non-flat spectra (i.e., tone-like signals), values close to one represent spectra which are flat (i.e., noise-like signals). It is somewhat related to spectral entropy which is a measure of randomness of a spectrum. In spectral entropy, the power spectrum is expressed as a probability distribution. If density values are more or less equal across frequency bins (i.e., the spectrum is flat), the randomness is high so it is the entropy.

Features like spectral flatness and spectral entropy are used in discrimination between music and non-music because music exhibits structure since it is purposely composed of harmonies and melodies. In my experiments, spectral flatness showed better performance than spectral entropy, especially in the case of noisy recordings.

1.3 Normalization

Before features are passed into the next stages of the pipeline, three normalization steps are applied to them in order to improve the performance of the model. This phase is not mandatory in the process, however, it is very important and helpful.

Outlier Removal

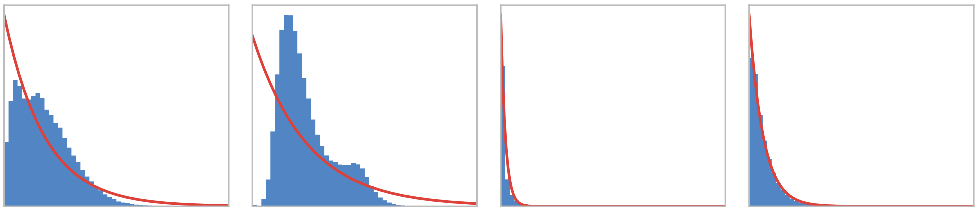
Extracted features can contain some values which significantly differ from the other values. These are called *outliers* [24]. Such points can occur for different reasons, but they are often related to the audio recording and/or processing circumstances, for instance, when a part of the audio signal is completely muted, then the value of spectral flatness is extremely high.

The classification process can suffer from the presence of outliers, however, their detection and removal are problematic. When the method is too eager, it can significantly affect the distribution of the data and consequently the classification performance.

I utilize a well-known technique, so-called *empirical rule* [24], which states that if the distribution is approximately bell-shaped, then 68% of the data fall into the interval $\mu \pm \sigma$, 95% into the interval $\mu \pm 2\sigma$ and 99.5% into the interval $\mu \pm 3\sigma$, where μ denotes the mean and σ denotes the standard deviation.

However, as can be seen in Figure 1.6, except for spectral centroid, all features are rather right-skewed than bell-shaped. Therefore, I apply the assumption about covering 99.5% of the data to the exponential distribution, which is a well-known right-skewed long tail distribution.

Recall that the probability density function for the exponential distribution is $p(x) = \lambda e^{-\lambda x}$ for $x \geq 0$, while 0 otherwise. Its mean is given by $\mu = \frac{1}{\lambda}$. The threshold θ of the data, where all samples above this threshold are marked as



(a) RMS **(b)** Centroid **(c)** Flux **(d)** Flatness

Figure 1.6: Distributions of features in a concert, where horizontal axis is for feature's value and vertical axis for count of frames in a bin. For spectral flux and spectral flatness, they resemble exponential distribution (red line). In the case of root-mean-square energy, the distribution is close to exponential, however, it is not always the case and it depends on the concert.

outliers, is computed as follows:

$$\begin{aligned} \int_0^\theta \lambda e^{-\lambda x}\, \mathrm{d}x &= 0.995\,, \\ 1 - e^{-\lambda\theta} &= 0.995\,, \\ e^{-\lambda\theta} &= 0.005\,, \\ -\lambda\theta &= \ln(0.005)\,, \\ \theta &= -\mu \ln(0.005)\,. \end{aligned} \tag{1.22}$$

Outliers are not actually removed from the data, but their values are cut down to the threshold value, in order to keep the continuity of the sequence of feature values. Moreover, I do not need them to be removed, the information that they are still very high is sufficient. For root-mean-square energy, spectral flux, and spectral flatness, the following conditional adjustment is applied:

$$\hat{x}_i^{\text{out}} = \min(x_i, \theta) \tag{1.23}$$

where x_i is a value from the feature series and θ is the threshold given by Equation (1.22). Note that even when a feature has bell-shaped distribution, outlier removal described above can be applied without any damage, since the mean of such distribution would be somewhere in the middle and hence the threshold would lay far on the right.

Moving Average

Two-sided *moving average* (sometimes called running mean or rolling average) is used to smooth the series in order to estimate the underlying trend [25]. Large variations could harm threshold-based techniques for segmentation and classification [10], as the curve is very noisy. Note that moving median, which is robust to outliers, could be used instead of moving average. However, as outlier removal described above is applied before the smoothing, moving median would not be that useful in this case.

The formula for computing moving average is

$$\hat{x}_i^{\text{avg}} = \frac{1}{\ell + 1} \sum_{j=-\frac{\ell}{2}}^{\frac{\ell}{2}} x_{i+j} \tag{1.24}$$

for $i = \frac{\ell}{2}, \frac{\ell}{2} + 1, \ldots, N - \frac{\ell}{2} - 1$, where ℓ is the window length over which the value is averaged, and is chosen to be an even number. Undefined $\hat{x}_i^{\text{avg}}$ at the ends of the series are not very important if a reasonably small ℓ is used, so when an arbitrary value is assigned, it does not distort the series considerably.

Min-Max Scaling

For the most multidimensional distance-based classification algorithms, scaling of individual features is important, since the difference in magnitudes can hurt their performance. Also, for threshold-based solutions, unified range allows using fixed thresholds regardless of original values of the series.

Min-max normalization is a simple linear transformation, preserving the relationship among original data, used to fit the values into a specific range $[\alpha, \beta]$ [26]. It is defined as

$$\hat{x}_i^{\text{scale}} = \frac{x_i - \min X}{\max X - \min X} \cdot (\beta - \alpha) + \alpha\,. \tag{1.25}$$

where $x_i \in X$ and X is the set of values which is to be scaled. The range $[\alpha, \beta]$ is usually chosen to be $[0, 1]$.

1.4 Segmentation

Many approaches have been introduced for automatic audio signal segmentation. In general, these techniques fall into two basic categories [9]:

- *Distance-based* (or *metric-based*) – In this method, chosen distance is computed between two adjacent feature vectors, or more often between two adjacent windows of such feature vectors. The peaks (i.e., local maxima) identified in the distance curve, which is derived by applying the distance function, then corresponds to the boundaries in the audio signal. This method is unsupervised and does not require any sort of training data, however, this fact implies that it cannot be used for labeling found segments.

- *Model-based* – This technique uses classification algorithms, which are trained on ground truth data, and then used to classify each audio frame independently. Either one universal model is used for all audio classes, or for each class there is an individual model. After frames are labeled, those adjacent ones, which belong to the same audio class, are merged together. The result of this approach is both audio segmentation and label for each segment.

In this thesis, I treat the song segmentation as the unsupervised problem. Therefore, the model-based approach cannot be the choice.

For the rest of this section, let $X = (x_n)_{n=0}^{N-1}$ be a sequence of frame-based feature vectors, where $x_n \in \mathbb{R}^d$ and d is the number of features. In this thesis, the features are described in Section 1.2. The frames are assumed to be independent, which is naturally not the case in the real world, but this fact is usually ignored in practice.

Most distance-based techniques formulate the boundary detection as follows [6]:

(1) either, there is no significant change at a time b, in other words, the whole sequence X falls into the same audio class;

(2) or there does exist such a change, that said, frames on the left side of the boundary at a time b fall into one category, whereas frames on the right side fall into another.

For that purpose, let X_A, X_B to be two adjacent windows from X split at time b, that is, $X_A = (x_n)_{n=0}^{b-1}$, $X_B = (x_n)_{n=b}^{N-1}$.

It is common practice to model the sequences of feature vectors as multivariate Gaussian distributions [6]. The cases of the problem listed above are then transformed into two assumptions: (1) the whole sequence is generated by a single multivariate normal distribution, that is, $X \sim \mathcal{N}(\mu, \Sigma)$; or (2) the windows are generated by two different multivariate normal distributions, that is, $X_A \sim \mathcal{N}(\mu_A, \Sigma_A)$ and $X_B \sim \mathcal{N}(\mu_B, \Sigma_B)$.

Outlier removal and moving average normalization methods, described in Section 1.3, are applied also on distance curves computed by the technique defined below.

Bayesian Information Criterion

The most popular distance measure used in audio segmentation, especially in the field of speech processing and tasks like speaker diarization, is *Bayesian information criterion* ($\Delta\,$BIC) [5]. The task can be converted into a model selection problem between the following models:

$$\begin{aligned} M_0 :&\quad X \sim \mathcal{N}(\mu, \Sigma) \\ M_1 :&\quad X_A \sim \mathcal{N}(\mu_A, \Sigma_A); X_B \sim \mathcal{N}(\mu_B, \Sigma_B) \end{aligned} \tag{1.26}$$

Bayesian information criterion of a model takes the form of a penalized log-likelihood and is defined as

$$\mathrm{BIC}_M = \ln \mathcal{L}(X|M) - \lambda \frac{|M|}{2} \ln N \,, \tag{1.27}$$

where $\mathcal{L}(X|M)$ denotes the maximum likelihood of data X given a model M, λ is the penalty weight, $|M|$ denotes the number of free parameters in a model M, while N denotes the length of the sequence X. Note that for the case of Gaussian model, $|M| = d + \frac{1}{2}d(d+1)$. In the original form of the Equation (1.27), the penalty λ was not present. It was introduced later as a "tuning" parameter for the segmentation performance [6].

Determining which one of these two assumptions is more likely corresponds to finding for which model the expression in Equation (1.27) is larger. For models M_0, M_1, their corresponding BIC formulas are

$$\mathrm{BIC}_{M_0} = \ln p(X; \hat{\mu}, \hat{\Sigma}) - \lambda \frac{|M|}{2} \ln N \,, \tag{1.28}$$

$$\mathrm{BIC}_{M_1} = \ln p(X_A; \hat{\mu}_A, \hat{\Sigma}_A) + \ln p(X_B; \hat{\mu}_B, \hat{\Sigma}_B) - \lambda \frac{|M|}{2} \ln N \,, \tag{1.29}$$

where $p(X; \hat{\mu}, \hat{\Sigma})$ denotes probability density function of the normal distribution and $\hat{\mu}, \hat{\Sigma}$ are maximum likelihood estimators for its mean vector and covariance matrix, respectively.

The difference between BIC_{M_0} and BIC_{M_1} is the distance metric used in boundary detection and is computed as

$$\Delta\,\mathrm{BIC}(b) = \mathrm{BIC}_{M_1} - \mathrm{BIC}_{M_0} \,, \tag{1.30}$$

where $b = 0, 1, \dots, N-1$.

If $\max_b \Delta\,\mathrm{BIC}(b) > 0$, then model M_1 fits the data better than M_0, that is, subsequences X_A, X_B are better estimated by parameters different for each subsequence than by parameters which are common for the whole window. The boundary is chosen to be where the value of $\Delta\,\mathrm{BIC}$ is maximal, i.e., where the peak of the distance curve is (as illustrated in Figure 1.7).

In order to detect multiple change points, the following algorithm was proposed [5]:

(1) Choose a small window starting at the beginning of the audio and denote it as the current window.

(2) Inside the current window, evaluate the test defined in Equation (1.30) for every sample b in the window.

(3) If no boundary was found, extend the current window. If a boundary does exist, set the current window to be a new window with the initial size and starting at the detected boundary.

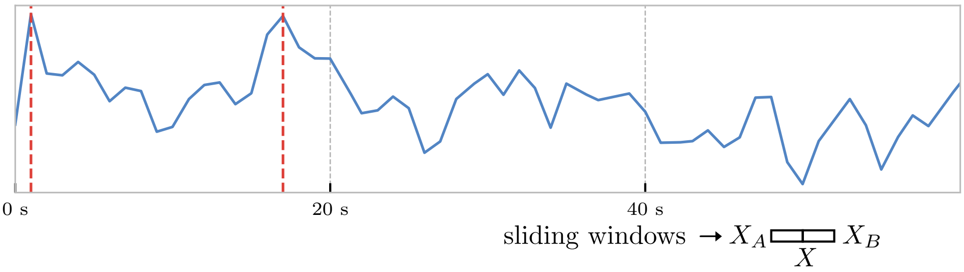


Figure 1.7: Distance curve with identified peaks (red dashed lines).

(4) Unless the end of the audio is reached, go to the point 2.

However, this approach is very computationally expensive as it has quadratic time complexity. There have been proposed many algorithm variants for improving the speed as well as the accuracy. One family of such improvements are multi-stage approaches [5], where faster, yet not so robust stages are performed before BIC stage, which then only refines previously found boundaries.

The second approach is computing the distance between constant-sized sliding windows [27]. In this case, $b := \frac{N}{2}$ where N is chosen to be an even number and the technique of growing N is not used. In this case, the penalty term in Equation (1.27) becomes constant (as the length of X is fixed) and therefore serves only as some threshold value. The Δ BIC then reduces to the log-likelihood ratio (also referred to as generalized likelihood ratio) [6]. Since full concerts are usually quite long, the computation of Δ BIC described above would be unbearable, so this simplified variant is used in this thesis.

Log-Likelihood Ratio

Likelihood ratio test is a statistical test where the problem is divided into two competing models/hypotheses H_0, H_1. The test is based on the ratio of likelihoods corresponding to these hypotheses, that is:

$$\Lambda(X) = \frac{\mathcal{L}(X|H_0)}{\mathcal{L}(X|H_1)}\,, \tag{1.31}$$

where $\mathcal{L}(X|H)$ denotes the maximum likelihood of data X under a hypothesis H. A decision to “reject H_0 in favor of H_1” is then made if the ratio Λ is less than some particular threshold.

Taking the models definition in Equation (1.26) and changing the notation to be $H_0 := M_0$ and $H_1 := M_1$, the likelihood ratio for the audio signal segmentation looks as follows:

$$\Lambda(X) = \frac{p(X; \hat{\mu}, \hat{\Sigma})}{p(X_A; \hat{\mu}_A, \hat{\Sigma}_A)p(X_B; \hat{\mu}_B, \hat{\Sigma}_B)}\,. \tag{1.32}$$

However, it is more practical to take the logarithm of the ratio for computational reasons. Furthermore, to be compatible with BIC framework described above, hypotheses are usually flipped so it can be also referred to as peak detection process (instead of valley detection). The equation for the *log-likelihood ratio* (LLR) then becomes

$$\mathrm{LLR} = p(X_A; \hat{\mu}_A, \hat{\Sigma}_A) + p(X_B; \hat{\mu}_B, \hat{\Sigma}_B) - p(X; \hat{\mu}, \hat{\Sigma}) \,. \tag{1.33}$$

Peak Detection

Simple, yet flexible peak detection method, described in [28], is used in this thesis. Let $(x_n)_{n=0}^{N-1}$ be a vector of interest, then x_n is selected as a peak if the following conditions hold true:

(1) $x_n = \max(x_{n-\ell_1}, \ldots, x_{n+\ell_2})$

(2) $x_n \geq \mathrm{mean}(x_{n-\ell_3}, \ldots, x_{n+\ell_4}) + \delta$

(3) $n - n' > \ell_5$

where $\ell_1, \ldots, \ell_5$ are tunable parameters, δ is a fixed threshold and n' denotes the last detected peak. The first condition finds a local maximum, the second one determines if the peak is sufficiently significant, and the third condition can prevent oversegmentation. In order to be adaptable to given data, I set δ to be equal to the standard deviation of data scaled by a parameter γ. This parameter is estimated empirically and is fixed.

1.5 Classification

Since the task of this thesis is defined to be unsupervised, I can utilize none of those classification algorithms used in the machine learning field, which rely on learning on a labeled dataset. Furthermore, clustering, which is a typical unsupervised approach, cannot be used either, because it basically groups points in a multidimensional space, and as can be seen in Figure 1.8, there is no clear separation in the space of features used in this thesis.

Three classification methods are introduced in this section – two of them being based solely on prior knowledge about the features, and one being taken from machine learning field. They can be applied separately, or combined together into a so-called ensemble.

The first approach implements arguably the simplest intuition that comes to mind when the problem of song segmentation occurs – the detection of silence; the second one utilizes the prior knowledge about features described in Section 1.2 and is basically threshold-based; and the last method is some kind of anomaly detection in the space of features.

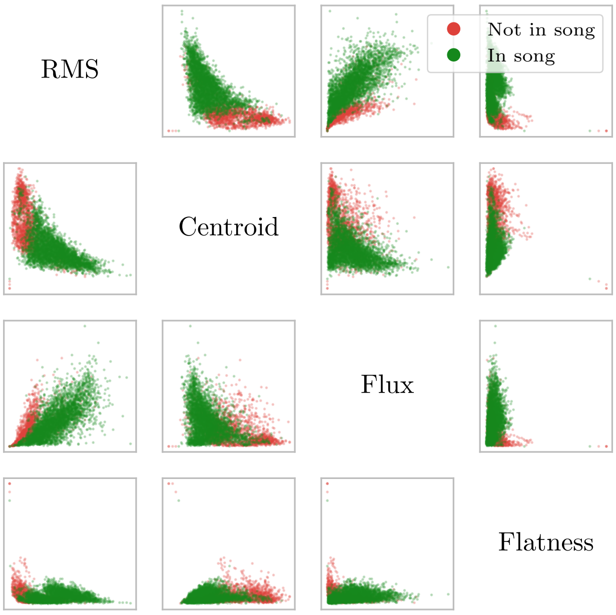


Figure 1.8: Scatterplot matrix of features in a concert. Features are plotted against each other (the boxes in the upper right half are mirrored in the lower left half). It is useful for visual analysis of correlation and class discrimination power of variables.

Classification is performed on frames combined into one-second data points aggregated using mean along the time axis, that is, a data point is composed of mean values of individual features within the one-second window. However, labels are assigned to the whole segments identified in the segmentation stage. The audio class is decided simply by the majority rule: if more frames in a segment are classified as *in song* than *not in song*, then the whole segment is labeled as *in song*, and vice versa. This solution is to some extent robust to odd frames which are assigned to an incorrect class because the majority of correct frames outweigh them in the segment.

The reason why a separate segmentation step is performed, instead of dividing the concert using just frame-based labelings, is that the latter would exhibit significant oversegmentation. The approach described in Section 1.4 identifies a boundary only where a considerable change between two windows occurs.

Silence Detection

Silence detection is very important and common in automatic speech processing, where it is usually referred to as *voice activity detection* or *speech activity detection*, as the presence of non-speech parts considerably affects the performance of models used in such tasks. Many techniques for voice activity detection have been proposed in the past (see [29] for their review). The most straightforward approach is based on energy values, where an empirical threshold is adopted. Despite its simplicity, it is widely used and achieves satisfactory results [29].

Although the full-concert domain is considerably different from the speech domain, I suppose that an energy-based technique can be used in this task. The key is in proper determination of the threshold. The approach for computing its value used in this thesis is as follows:

$$\theta = \gamma \cdot \mathrm{mean}(X^{\mathrm{RMS}})\,, \tag{1.34}$$

where X^{RMS} is the sequence of root-mean-square energy values for the full concert given by Equation (1.16), and γ is the parameter which should be estimated empirically. In other words, it is just a scaled mean of the energy across frames.

All frames with root-mean-square energy below this threshold are then labeled as *not in song*, whereas the rest is labeled as *in song*.

Rule-Based Classification

Assuming the properties of features described in Section 1.2, it is possible to build a threshold-based classifier that labels frames according to their spectral characteristics. Here is the concise list specifying what values individual features exhibit, considering the task addressed in this thesis:

- *Root-mean-square energy* – as music is the fundamental part of the full-concert content, it is assumed that the root-mean-square energy, which strongly correlates with the loudness, is higher for songs than for parts between them.

- *Spectral centroid* – applause and similar noise-like content exhibit higher centroid than is the case for music.

- *Spectral flux* – as music spectrum is more structured, its value of spectral flux is usually higher.

- *Spectral flatness* – applause spectrum is usually flat, therefore spectral flatness of music is much lower than for parts between songs.

As can be seen in Figure 1.9, in the ideal case, a feature curve looks like a hill or a valley – depending on the property of the feature – in parts between songs. As with silence detection, the problem here is the determination of the threshold.

Generally, two techniques can be applied: global (fixed) or local (adaptive) thresholding. In fixed setting, there is one threshold for all frames in the whole

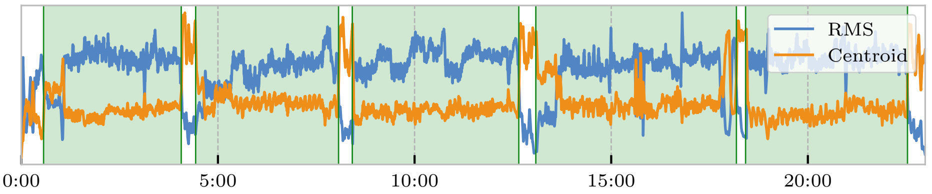


Figure 1.9: Values of root-mean-square energy and spectral centroid for a part of a concert. Green blocks represent songs.

concert. This is the more straightforward approach, however, its assumption is that the audio characteristics remain the same for the whole duration of the concert.

On the other hand, adaptive thresholding is able to handle significant audio changes, caused by for example location moves of the recording author or technical difficulties. Furthermore, it should exhibit superior performance in hill/valley identification, since it considers only the neighborhood of it.

I take the arithmetic mean of a sequence as the base for the threshold, which is then weighted by an empirically estimated parameter γ. I argue that the mean is suitable statistic here because its value is biased toward the mass of the data (in this case, music content) and therefore is able to reveal – in some form – unusual segments (in this case, parts between songs). Note that the parameter γ can be estimated for each feature separately in order to fine-tune the model for better performance. On the other hand, this is also a weakness, because the success rate of the model is strongly affected by the parameters' values – which moreover depend on a particular audio signal being processed –, and it is quite prone to poor settings.

The threshold-based method also offers some kind of confidence score – basically the difference between the value and threshold. This score can be used to weigh the class estimated by individual features in order to scale the amount of confidence in the final prediction.

Let $W_n(X^{(f)})$ be a function returning the neighboring frames of n-th frame in $X^{(f)}$, which denotes the curve for feature f. It is given by

$$\begin{aligned} W_n(X^{(f)}) &= (x_i)_{i=a}^{b}\,, \\ a &= \max\left(n - \frac{\ell}{2}, 0\right)\,, \\ b &= \min\left(n + \frac{\ell}{2}, N-1\right)\,, \end{aligned} \tag{1.35}$$

where ℓ denotes the length of the window and is chosen to be an even number. Note that for fixed thresholding approach, $W_n(X^{(f)})$ is just an identity function.

Let $\theta_n^{(f)}$ be the threshold value for n-th frame in feature f computed as

$$\theta_n^{(f)} = \gamma^{(f)} \cdot \mathrm{mean}\left(W_n(X^{(f)})\right) \; , \tag{1.36}$$

where $\gamma^{(f)}$ denotes the threshold scaling parameter for feature f.

Given the threshold $\theta_n^{(f)}$, let $g(x_n^{(f)}, \theta_n^{(f)})$ be a function, whose sign corresponds to the estimated audio class, and whose value represents the amount of confidence of such classification. The body of the function depends on the properties listed at the beginning of this subsection, and is defined as follows:

$$g(x_n^{(f)}, \theta_n^{(f)}) = \begin{cases} x_n^{(f)} - \theta_n^{(f)} & \text{for } RMS, Flux \\ -(x_n^{(f)} - \theta_n^{(f)}) & \text{for } Centroid, Flatness \end{cases} \tag{1.37}$$

Finally, the result estimation is given by the weighted contribution of all features:

$$y_n = \sum_f \left| g(x_n^{(f)}, \theta_n^{(f)}) \right| \cdot \mathrm{sgn}\left(g(x_n^{(f)}, \theta_n^{(f)}) \right) \; . \tag{1.38}$$

Frames with $y_n > 0$ are classified as *in song* and the other frames are given the *not in song* label.

Anomaly Detection

Anomalies (or outliers) are data points which significantly deviate from the remaining data. In other words, they do not conform to a normal behavior, whose definition is based on the application. Anomaly detection is applicable in various fields, for example, detection of intrusion in computer systems, detection of financial fraud, interesting sensor events or medical diagnosis [30].

Techniques that operate in the unsupervised mode assume that normal instances are far more frequent than anomalies [31]. I take this assumption as to some extent valid for the task since songs usually fill much more content than parts between them. Although there seem to be no obvious clusters in the feature space (Figure 1.8), I suppose that an anomaly detection can be partially successful in the discovery of frames which are likely to be from parts between songs.

There are several categories of anomaly detection approaches with many algorithms in them (for a review, see [31] or [30]). Distance-based techniques incorporating nearest neighbor analysis are widely used, as they are very straightforward and interpretable. They work on the assumption that normal instances occur in dense neighborhoods, while anomalies lay far from their closest neighbors. See Figure 1.10 for the illustration of this concept.

A basic approach is to define the anomaly score of a data point as its distance to its k-th nearest neighbor. Value of k serves as a parameter of the model which needs to be estimated from data. Small sets of anomalies are discoverable if this parameter is greater than one [30]. Usually, applying

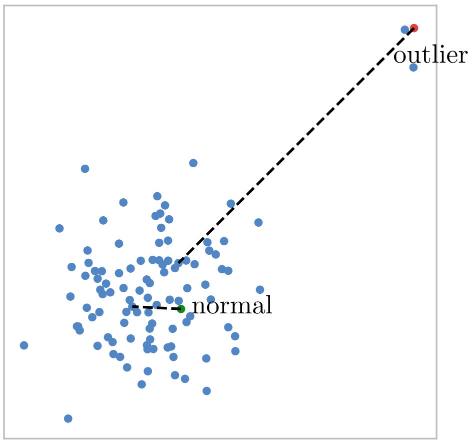


Figure 1.10: Illustration of nearest neighbor anomaly detection. The distance of k-th neighbor of outlier is much bigger than that of normal point.

some threshold value on the score is used in order to determine if the point is anomalous or not. Several variants of this method have been proposed (see [31] for their description). However, I use this simple approach, because it exhibits better performance in my experiments than some of those extensions.

The requirement of the nearest neighbor technique is the specification of a distance or similarity measure between two points. The most popular one is Euclidean distance, but others can be used and the choice depends heavily on the application domain [31].

After the anomaly score curve is computed, I basically use the same techniques described in the previous subsection about the rule-based classifier, that is, the window function is defined as in Equation (1.35), and the threshold value is computed as in Equation (1.36). All scores d_n which are below the threshold θ_n are considered as *in song* and scores above as *not in song*.

One of the advantages of this approach is that it does not rely on any prior hardcoded knowledge about the features. Its parameters are also simple to estimate, contrary to the rule-based model where the number of parameters is problematic.

Ensemble

An *ensemble* is a set of models which combines in some way their decisions[9] to obtain more accurate predictions. Empirical studies have shown that ensembles are superior to single models, and some theoretical studies attempt to explain this success [32].

[9] Or their learning algorithms, or different views of data, or other specific characteristics.

There are two main categories of ensemble techniques. *Non-generative* try to combine existing base classifiers, whereas *generative* methods generate new base learners in a way which improves their diversity and accuracy [32]. Since classification models presented in this thesis are designed to work on the whole audio of the concert, feature space is quite low-dimensional and adapting the task to generative approaches would require further analysis of data and models, the non-generative approach, which is considerably simpler and easier to understand, is chosen for this thesis.

The most popular and straightforward method is the *majority voting* ensemble [32]. In this case, "votes" represented by base models' predictions are collected and the class which receives the majority is decided to be the final prediction. Since three classifiers are used and the problem has two classes, in this thesis, majority voting has always a clear decision.

1.6 Postprocessing

After segmentation and classification stages are completed, there are often segments, labeled as being from the positive class, which are presumably not actual songs considering their short length. Although exceptions exist[10], musical compositions last usually tens of seconds at minimum.

For that reason, song segments which are shorter than a threshold T are either merged with a neighboring positive segment or discarded, according to the following rules:

- If the distance to the closer neighbor is less than a threshold C, then the segment is merged with this neighbor.
- Otherwise, the segment is discarded.

[10] For an extreme example, the song "You Suffer" by the British band Napalm Death is only 1.3 seconds long.

Chapter **2**

Experimental Evaluation

This chapter first introduces the dataset and performance metrics, together serving as the evaluation framework of the thesis. Then, the implementation of the system proposed above is briefly described. At the end of this chapter, the results are presented and the influence of different parameters is evaluated.

2.1 Data and Metrics

2.1.1 Dataset

As far as I know, there is no publicly available dataset of song time annotations in full concerts other than that used in "Set list identification" task in MIREX[11]. However, the dataset contains only precomputed chroma features[12] and not the original audio files. For that reason, it cannot be used for evaluating model presented in this thesis since it is based on different features.

Furthermore, in publicly released part of the set, there are only three artists, therefore the genre range is very limited. Also, the audio quality of recordings is not known, so it could not be used for noise robustness evaluation.

For the purposes of this thesis, I collected a dataset of full concerts from video-sharing website YouTube, along with their high-quality song time annotations, usually produced by myself. It features several artists playing various musical genres, and the audio quality is purposely diversified, addressing the shortcomings of the MIREX dataset.

The dataset is available on GitHub[13], which is a contribution platform so the dataset can be extended in the future, either by myself or by the

[11] `http://www.music-ir.org/mirex/wiki/2017:Set_List_Identification`

[12] Chroma-based audio features represent the signal spectrum as twelve (or multiply of twelve) pooled bins corresponding to twelve pitch classes of the equal-tempered scale, used in western music [33].

[13] `https://github.com/pnevyk/time-annotations-of-live-concerts-dataset`

community. For results presented in this thesis, the version from 1th May 2018[14] is used.

To comply with the copyright rules, only links to the content-sharing service is present in the repository. If, for any reason, a concert is removed from its destination, it will not be available in the dataset anymore. However, it is a trade-off which on the contrary allows collecting a larger dataset more conveniently without breaking the copyright laws.

Table 2.1 lists all concerts available at the time of writing this thesis, along with other metadata.

2.1.2 Metrics

F-measure and Specificity

Although the most common scoring measure for a model evaluation is accuracy, given as the ratio of correctly classified data points and all data points, in the case of imbalanced datasets, it can lead to misleading results [34]. Imbalance occurs when one class is significantly underrepresented in a dataset. Considering the amount of song content in comparison with non-music content, this disproportion is indeed the case for this thesis. If a model labels all frames as *in song*, then it still gets quite a good accuracy score.

For this reason, *f-measure* (also f-score or f_1 score) is employed, as it balances the *precision* (model's certainty in identification of the positive class[15]) and *recall* (model's capability of the positive class discovery).

More precisely, all the following measures derive their values from *confusion matrix* (illustrated in Figure 2.1), which expresses all four cases that can occur in binary classification scenario:

- true positive (TP) – model's prediction of the positive class corresponds to the ground truth,
- false positive (FP) – model incorrectly classified an instance as positive,
- false negative (FN) – model failed to identify a positive instance,
- true negative (TN) – model's prediction of the negative class corresponds to the ground truth.

[14] It is tagged as `v1.0.0` in the relevant repository.

[15] "Positive" here refers to one of two classes which is chosen arbitrarily to be a positive case, for example *in song* class in this thesis.

Concert	No. of songs	Genre	Audio quality
AC/DC – Capital Centre 1981	18	Hard Rock	Bad
AC/DC – Munich 2001	21	Hard Rock	Good
Adele – Royal Albert Hall 2011	17	Pop	Good
B.B. King – A Blues Session 1987	12	Blues	Average
Beatles – Atlanta Stadium 1965	11	Rock and Roll	Bad
CHVRCHES – Glastonbury 2016	12	Synth-pop	Good
Code Orange – The Electric Factory 2014	7	Hardcore	Good
Coldplay – Toronto 2006	18	Pop Rock	Good
Eminem – Reading Festival 2017	32	Hip Hop	Good
Katy Perry – Big Weekend 2017	11	Pop	Good
Metallica – Moscow 1991	14	Heavy Metal	Good
Michael Jackson – Rome 1988	12	Pop	Average
Nails – This is Hardcore 2013	10	Hardcore	Good
Punch – The First Unitarian Church 2011	13	Hardcore	Average
System of a Down – Lowlands 2001	16	Alternative Metal	Average
Wu-Tang Clan – Hultsfreds Festival 1997	11	Hip Hop	Average
	235		

Table 2.1: List of concerts with their metadata.

	Ground truth positive	Ground truth negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Figure 2.1: Confusion matrix illustration.

Given these values, precision, recall and afterward f-measure is computed as follows:

$$\text{Precision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}}\,, \tag{2.1}$$

$$\text{Recall} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}}\,, \tag{2.2}$$

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}\,. \tag{2.3}$$

Note that f-measure is, in fact, the harmonic mean of precision and recall. In a more general variant, their contribution can be weighted by some parameter to put emphasis on model's focus (for example on the ability to identify negative cases in an imbalanced dataset). However, I do not take this approach in this thesis.

Another important statistic for the problem of this thesis is *specificity* – the rate of correctly identified instances of the negative class. If a model classifies all frames as *in song*, then it still achieves considerably high f-measure, but the specificity – in other words, identification of *not in song* parts – is zero. Such model would not be very useful in practice for this application. The formula for specificity takes this form:

$$\text{Specificity} = \frac{\#\text{TN}}{\#\text{TN} + \#\text{FP}}\,. \tag{2.4}$$

As model presented in this thesis performs classification on the one-second basis, these statistics are computed having second-long labeled frame aggregations as instances. This measurement does not take the segmentation into account at all (or at least not explicitly), it is rather an evaluation of the classification only.

Normalized Dynamic Time Warping Error

There are multiple approaches for measuring segmentation accuracy since audio segmentation is a very common task in signal analysis. Nevertheless, the problem addressed in this thesis is a little bit specific and different from traditional segmentation tasks.

One scoring measure could be the application of f-measure (as described above) on segment boundaries. This approach is used for example in [35]. The problem is that the song boundaries in live performances are often not clear and their identification is not trivial, hence every scoring method which evaluates only the fact if a boundary was identified precisely or not, will result in a poor performance and not be very helpful accuracy indication.

Segmentation error score (SER) is another way of evaluating segmentation task, and it is defined also for multilabel[16] tasks. SER is computed as a fraction of the class time that is not correctly assigned [36]. The signal is divided into segments where the boundary of every segment is a class change point, either in the model's prediction or in the ground truth. For each segment, the number of incorrectly assigned classes is weighted by the segment's length, and the summation of these error values is then normalized with the summation of every segment's length multiplied by the number of ground truth labels in it.

However, this error measure is not quite suitable for the task addressed in this thesis, whose application is to provide the user with time boundaries of songs. If a model produces several very short segments in a song, while labeling them as *not in song*, then its SER is still small because of shortness of these segments. But because of these boundary false positives, the usability of such model is not proportionate to the low level of its segmentation error.

In [14], where authors address a very similar problem to mine, for segmentation performance evaluation a measure called *boundary deviation* is used. It is computed as absolute values of differences between predicted the boundaries and the ground truth, but only of songs which were correctly labeled (song name identification is an integral part of their task). As there is no such concept of song recognition in my thesis, I cannot take this measure in this form.

Nevertheless, taking inspiration from it, I propose an error measure which I call *normalized dynamic time warping error*. The cornerstone of the error measure is *dynamic time warping* (DTW), a well-known and popular algorithm for time series similarity computation [37]. It minimizes the effects of shifting and distortion in time, and also works with sequences of different length.

The requirement of DTW is the specification of a distance measure between two points of time series. The choice is crucial and depends on the actual application. The algorithm then builds a distance matrix representing pairwise distances between the series, and finds the optimal path in it (so-called *align-*

[16] In multilabel classification, more than one class can be assigned to an instance. It is a different term from multiclass classification, where there are multiple possible classes, but just one can be assigned to an instance.

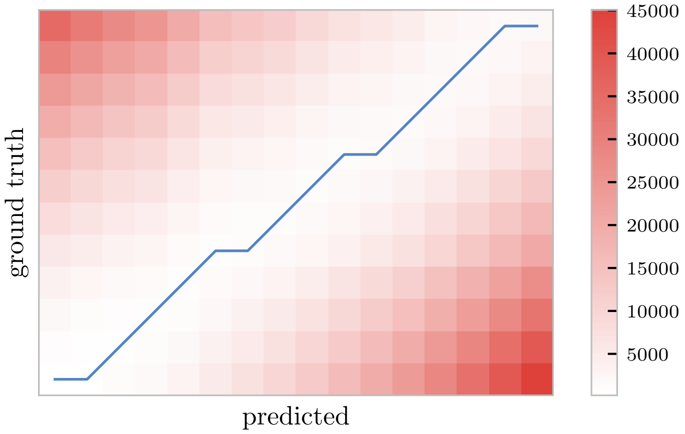


Figure 2.2: Illustration of dynamic time warping algorithm.

ment path) regarding these three conditions:

(1) the starting and ending points of the path must be the first and the last points of sequences,

(2) time order of the path must be preserved, that said, it never goes back in time in either series,

(3) the step size is always one, either horizontal, diagonal or vertical way.

See Figure 2.2 for an example of a distance matrix and corresponding alignment path.

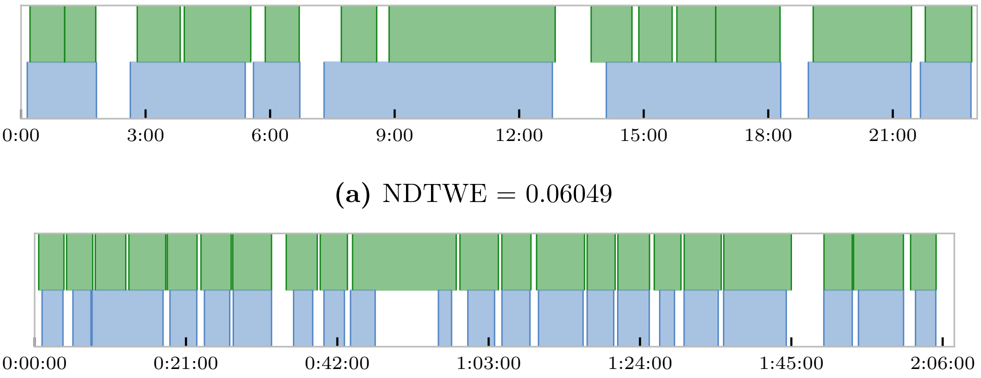
Let $S_{\text{mod}} = (s_i)_{i=1}^{N}$ be a sequence of pairs $s_i \in (\text{start}, \text{end})$ representing the song boundaries identified by a model, and $S_{\text{ref}} = (s_i)_{i=1}^{M}$ be a sequence of the ground truth boundaries. Taking the inspiration from the boundary detection evaluation used in [14], I use boundary deviation in DTW framework to get the segmentation error of a model.

Two distances, defined in Equations (2.5) and (2.6), are employed: the first measures the difference between both start and end boundaries, while the second takes only the start boundary into account. The rationale behind the latter distance is that only start times of songs are usually requested and provided on full-concert-sharing services since the main purpose of the time annotations for the listeners is being able to find a song of interest.

$$d_{\text{both}}(a, b) = |a_1 - b_1| + |a_2 - b_2| \tag{2.5}$$

$$d_{\text{start}}(a, b) = |a_1 - b_1| \tag{2.6}$$

One problem is that the error value is to some extent dependent on the number of songs in the concert. If a model labels the whole audio signal as



(a) NDTWE = 0.06049

(b) NDTWE = 0.00895

Figure 2.3: Examples of songs segmentation results along with their NDTWE. Green segments represent the ground truth, blue segments correspond to the estimation.

in song, then the error is higher with a greater number of songs in ground truth and therefore has little descriptive power. For this reason, it should be somehow normalized so it is as independent on concert being classified as possible in order to be a descriptive measure regardless of what is actually evaluated.

I take the case when the whole audio signal is classified as *in song* as some form of the worst case, and its error as a normalization factor. The value of this normalization incorporates both the number of songs and the length of the concert.

Value of normalized dynamic time warping error is computed as follows:

$$\text{NDTWE} = \frac{\text{dtw}_d(S_{\text{mod}}, S_{\text{ref}})}{\text{dtw}_d\Big(\big((0, L)\big), S_{\text{ref}}\Big)}, \tag{2.7}$$

where d is either d_{both} or d_{start} and denotes the distance used in dynamic time warping algorithm, and L is the length of the concert in seconds.

Figure 2.3 shows two examples of segmentation results along with their computed normalized dynamic time warping error, in order to get an intuition about its behavior and properties.

One of the advantages of this error measure is that it penalizes both over- and undersegmentation (although the latter is penalized more than the former), which are both harmful for a practical application addressing this task. On the other hand, concerts which are longer and have a greater number of songs tend to have a smaller error, due to the way how the normalization factor is computed. Nevertheless, I argue that it is still a suitable measure.

2.2 Implementation

The system described in this thesis is implemented in Python programming language using scientific libraries from SciPy ecosystem [38], audio and music processing library Librosa [39], and machine learning toolkit Scikit-learn [40]. All these libraries provide a pythonic interface, but heavy computation parts are backed by performant implementations in C programming language.

All the code is released as an open source package and is available on GitHub[17]. The implementation is modular so it is easy to add a new technique. Interfaces of functions provide very convenient possibility to configure all the parameters inside the whole model from the top level, thus allowing to experiment with different values in order to achieve the best performance possible.

Values of parameters used in the model, of which results are presented in Table 2.6, were estimated using manual analysis and *grid search* algorithm, which basically evaluates all combinations from given parameter space. The main performance criterion was normalized dynamic time warping error described in Subsection 2.1.2.

Values of the sampling rate, spectrum window shift, and number of frequency bins of the spectrum were set to default values of Librosa library, that is, 22.05 kHz, 512 frames, and 84 and 128 for constant-Q transform and mel-scaled spectrum, respectively. Different values were experimented with, however, the default ones have shown the best performance.

For practical application, a segmentation tool must have a reasonable running time and memory consumption. Although it is not necessary – and considering the length of full concerts, it is perhaps impossible – to offer almost instant-time experience, the segmentation should take an acceptable amount of time even on low-cost personal computers.

The tool presented and implemented in this thesis takes roughly 30 seconds with 700 MB and 2 minutes with 3.5 GB on processing twenty-minute and two-hour long uncompressed audio file, respectively, on a middle-end computer, while approximately 75% and 20% of the time is spent in the spectrum computation and signal segmentation, respectively. Nevertheless, the runtime performance and memory efficiency were not the goals of this thesis, and there is room for optimization and parallelization.

2.3 Results

This section briefly presents results obtained during experimental evaluation. The most descriptive measure for the task is the error, since it is a score for both segmentation and classification jointly, while f-measure and specificity

[17] `https://github.com/pnevyk/segson`

Feature	Spectrum	Scale	F-measure	Specificity	Error
RMS	STFT	1.0	82.95%	78.50%	0.03699
RMS	CQT	0.9	87.20%	79.19%	0.03505
RMS	Mel	1.0	72.62%	82.06%	0.05569
Centroid	STFT	1.0	52.99%	44.43%	0.05346
Centroid	CQT	1.1	90.41%	67.77%	0.04043
Centroid	Mel	0.9	78.58%	84.40%	0.04103
Flux	STFT	0.9	82.70%	74.53%	0.03647
Flux	CQT	1.0	77.9%	81.95%	0.03681
Flux	Mel	0.9	78.03%	82.25%	0.04331
Flatness	STFT	1.0	61.51%	33.68%	0.05250
Flatness	CQT	1.0	59.90%	44.11%	0.05481
Flatness	Mel	0.8	70.70%	56.05%	0.07099

Table 2.2: Performance of individual features, where scale values are those that result in a minimal error.

	STFT	CQT	Mel
RMS	1.3	1.1	1.1
Centroid	1.3	1.1	1.3
Flux	1.1	1.1	1.1
Flatness	1.3	1.3	1.8

Table 2.3: Parameter values used in the rule-based model.

characterize the performance of the classification only. If not specified otherwise, all presented errors stand for normalized dynamic time warping error with d_{both} distance (Equation (2.5)).

The constant-Q transform proved to be the most suitable spectrum for this task, as can be observed throughout this section, where comparisons with other spectra occur. For this reason, if it is not specified which spectrum is used, it is always CQT.

The predictive power of individual features is listed in Table 2.2. It was measured incorporating the same technique as used in silence detection model. Note that scale parameters deviate from 1 less than in the rule-based model on average, where the features are applied jointly (parameter values used in the rule-based model are listed in Table 2.3).

It can be seen that root-mean-square energy is the best descriptor, followed by spectral flux. An important observation is that spectral centroid performs much better for logarithmic spectra than for linear STFT. All features, except for spectral centroid, have a significantly greater error in the case of the mel-scaled spectrum.

Peak-normalized spectral flux turned out to be a very poor feature for

Model	Spectrum	F-measure	Specificity	Error
Silence	STFT	82.95%	78.50%	0.03699
Silence	CQT	87.20%	79.19%	0.03505
Silence	Mel	72.62%	**82.06%**	0.05569
Rule-based	STFT	77.62%	71.34%	0.03701
Rule-based	CQT	86.35%	80.12%	**0.02962**
Rule-based	Mel	88.30%	71.10%	0.03919
Anomaly	STFT	82.18%	58.04%	0.03508
Anomaly	CQT	87.70%	75.94%	0.02986
Anomaly	Mel	86.26%	52.73%	0.04823
Ensemble	STFT	83.69%	76.83%	0.03692
Ensemble	CQT	**88.92%**	81.30%	0.02968
Ensemble	Mel	87.96%	75.20%	0.03896

Table 2.4: Results of various model and spectra combinations.

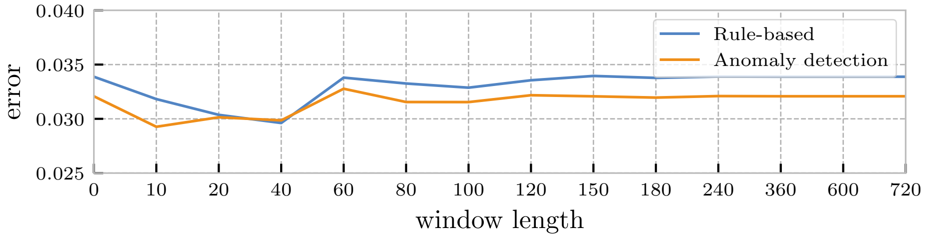


Figure 2.4: Model performance depending on the length of classification window. Zero indicates that no window is used.

discrimination between songs and parts between them. It implies that absolute contribution to flux computation plays an important role, and consequently that success of spectral flux is partially caused by its correlation with the loudness.

Table 2.4 shows the results for all combinations of models and spectra. The lowest error is achieved by the rule-based model, however, the difference from the ensemble model is negligible. On the other hand, values of f-measure and specificity are better in the ensemble model. Therefore, the latter is taken as the best model.

The influence of choosing the length of the window, in which threshold in rule-based and anomaly detection models is computed (from now on called "classification window"), is visualized in Figure 2.4. The length indeed plays some role in the performance, and as visible in the figure, its use improves the results.

As can be seen in Table 2.5, postprocessing stage has a negligible impact on overall performance. In the case of anomaly detection, the results are even

Model	Postprocessing	F-measure	Specificity	Error
Silence	×	87.45%	79.17%	0.03488
Silence	✓	87.20%	79.19%	0.03505
Rule-based	×	86.69%	79.92%	0.02988
Rule-based	✓	86.35%	80.12%	0.02962
Anomaly	×	88.00%	75.06%	0.03051
Anomaly	✓	87.70%	75.94%	0.02986
Ensemble	×	89.18%	81.28%	0.02977
Ensemble	✓	88.92%	81.30%	0.02968

Table 2.5: Influence of postprocessing.

slightly worse.

Summarization results of individual concerts are listed in Table 2.6. In this table, there are values of error for both introduced distances, that is, d_{both} and d_{start}. The latter is a valuable measure since usually only the start times of the songs are requested by the listeners. If this error is lower for a concert, it loosely indicates that the algorithm was more successful in detection of the start times, while the end times caused troubles, and vice versa.

Concert	F-measure	Specificity	Error (d_{both})	Error (d_{start})
AC/DC (Capital Centre)	81.81%	86.99%	0.02338	0.02004
AC/DC (Munich)	88.40%	93.58%	**0.00895**	0.00719
Adele	87.68%	86.23%	0.01092	0.01079
B.B. King	88.32%	96.94%	0.01176	0.01062
Beatles	91.47%	67.81%	0.03371	0.03059
CHVRCHES	88.68%	88.89%	0.02613	0.02882
Code Orange	94.61%	60.32%	0.03245	0.03586
Coldplay	80.08%	88.72%	0.01991	0.01866
Eminem	90.51%	82.76%	0.03308	0.03831
Katy Perry	88.14%	97.57%	0.02582	0.03125
Metallica	93.49%	65.49%	0.03672	0.02673
Michael Jackson	90.65%	76.81%	0.02192	0.02377
Nails	92.80%	73.51%	0.04996	0.04474
Punch	93.16%	64.81%	**0.06049**	0.05292
System of a Down	90.67%	89.22%	0.02286	0.02080
Wu-Tang Clan	82.26%	81.08%	0.05682	0.06786
	88.92% ± 4.16%	81.30% ± 11.51%	0.02968 ± 0.01490	0.02931 ± 0.01562

Table 2.6: Segmentation and classification results of all concerts using the best model.

Chapter 3

Discussion

A solution for the unsupervised song segmentation in full concert audio was presented throughout this thesis. Despite being used frequently in the literature, the application of described techniques in this particular domain is not so common, partly because full concerts are not so popular and extensively studied area in audio processing and machine learning communities. Therefore, there are some findings which deserve further discussion.

First, the results convincingly show that logarithmically scaled spectra (constant-Q transform and mel-scaled spectrum) exhibit better performance than linear short-time Fourier transform. This is not surprising since the main content of interest in this work is music which is based on logarithmic scales following the properties of human perception of sound.

Although features chosen for the system have, in theory, properties suitable for the task, the assumption about their behavior does not always hold true in the real world. And even in instances where they behave ideally, the threshold of deciding if a frame corresponds to a song or not is strongly dependent on particular audio signal conditions and semantic content as well. This causes serious problems, especially for models where the features' behavior is somehow hardcoded into. In this thesis, this applies to the rule-based model the most.

Segmentation using log-likelihood ratio proved to be sufficiently successful. Although it exhibits some degree of oversegmentation, this issue was to some extent solved by merging segments labeled as the same audio class and eventually the postprocessing stage.

Silence detection technique has shown very decent performance considering its simplicity. However, this probably indicates that other models, introduced in this work, did not bring the expected improvement. Identification of quiet parts is arguably the first solution which comes to the mind when this task is presented, and more complex approaches should significantly outperform it.

Rule-based model acts as some kind of extension of silence detection as it incorporates all features instead of just energy. The major weakness turned out to be the setting of its parameters. It is very prone to poor configuration and slightly changed values often lead to significantly different performance. That also means that values presented as the best in this work are optimized only for the evaluation dataset and the success rate for other concerts might be lower.

Nearest neighbor anomaly detection does not explicitly utilize any prior knowledge about the features. Despite that fact, it achieves decent results. Its performance could be slightly tuned by weighting the contribution of features by scaling to different ranges for each feature. In such case, the prior knowledge would be introduced into this model, however, this information would be only external – only the space passed into the model would differ, whereas the internals would remain exactly the same. Nevertheless, this improvement would be only minor and it would introduce the problem of setting the ranges, very similar to the thresholds' estimation in the rule-based technique.

Although it cannot be said confidently due to the limited size of the evaluation dataset, the solution does not seem to be significantly worse or better depending on the genre of a concert. The only exception is probably hip hop and related styles, where songs usually change without any pause. Furthermore, audio quality does not affect the performance considerably either, as the error in noisy concerts is comparable with professionally recorded ones.

3.1 Future Research

The most difficult and unexplored area of this thesis is the classification phase. As no extrinsic information is given to a classifier, none of the traditional algorithms can be employed. There are two ways how to achieve better performance in the pipeline presented in this work: either come up with different features which are better predictors of audio classes of interest, or adopt a better technique which is able to discover non-song data points in the space of features. Building a classifier based on hardcoded knowledge of features' properties is problematic and not flexible.

Although the segmentation phase performs quite well in this work, there is certainly room for improvement. Technique capable of identifying boundaries only at those times where a song actually starts or ends would significantly help in overall performance and effectively suppress the importance of the classification stage. However, such a method is arguably much more difficult to develop than a better classifier.

One noteworthy limitation of models proposed in this thesis is that they work only on very short time scale. However, there is a huge potential in temporal features which are able to characterize audio signal with long-term relationships since the music by definition is composed of melodies (sequences of notes) and has a rhythmic structure.

There are two aspects which more advanced models should deal with: immediately consecutive songs and breaks in them. In order to solve these problems, techniques from music similarity area should be employed. A robust solution would be able to distinguish two distinct songs even when there is almost no pause and to merge two segments if they belong to one song, solely based on the content of the segments identified as music.

Quite a different approach could be incorporating deep learning methods. Their great success in machine learning field leads to the motivation for their adoption in this task as well. One problem is that these techniques usually need a huge amount of data. However, collecting a large number of concert audio without time annotations for unsupervised learning is not that hard due to their abundance in today's world. Nevertheless, my initial experiments with neural networks have shown that their application is not trivial, and it would require thorough analysis and solid experience.

Conclusion

The goal of this thesis was to develop a system for the song segmentation in full concert audio which works in the unsupervised setting, that is, without labeled dataset for pretraining a segmentation or classification model. The reason for this constraint is that the solution is not biased and should to some extent function with various musical genres and audio qualities without collecting a massive ground truth data. This assumption was evaluated experimentally and the results were presented.

The whole pipeline for solving such problem was proposed, from sound and music description through spectrum computation and feature extraction to segmentation and classification. Then the experimental evaluation details were described and the results of the implementation were discussed. The implementation is released as an open-source Python package.

It turned out that the problem is non-trivial. There were two main difficulties: (1) the diversity of concerts and their spectral properties, and (2) the fact that audio signal of polyphonic music, especially if recorded in noisy environments, is complex and hard to retrieve the information from.

This thesis employed widely used spectrum types, features and segmentation technique. As far as I know, the literature lacks advanced methods for song/non-song classification without pretraining from a labeled dataset. Therefore I introduced three simple models and their ensemble. The score achieved on the dataset is 88.92% f-measure and 81.30% specificity of correctly labeled seconds in audio signal, and visualizations demonstrate that the song segmentation is relatively successful. Although it is not perfect, the best model from this thesis serves as an appropriate baseline for future approaches.

This work is perhaps the first attempt in this research area, which eventually results in a practical application that is able to discover the time boundaries of songs in full live concerts. Such application is useful in media-sharing services, where set lists of concerts are often requested by the users.

Bibliography

[1] Orio, N. Music Retrieval: A Tutorial and Review. *Foundations and Trends in Information Retrieval*, volume 1, no. 1, jan 2006: pp. 1–96, ISSN 1554-0669, doi:10.1561/1500000002. Available from: `http://dx.doi.org/10.1561/1500000002`

[2] Humphrey, E. J.; Bello, J. P.; et al. Feature learning and deep architectures: new directions for music informatics. *Journal of Intelligent Information Systems*, volume 41, 2013: pp. 461–481, doi:10.1007/s10844-013-0248-5. Available from: `https://doi.org/10.1007/s10844-013-0248-5`

[3] Kos, M.; Grasic, M.; et al. On-Line Speech/Music Segmentation for Broadcast News Domain. In *2009 16th International Conference on Systems, Signals and Image Processing*, 2009, pp. 1–4, doi:10.1109/IWSSIP.2009.5367789. Available from: `https://ieeexplore.ieee.org/document/5367789/`

[4] Pikrakis, A.; Giannakopoulos, T.; et al. A Speech/Music Discriminator of Radio Recordings Based on Dynamic Programming and Bayesian Networks. *IEEE Transactions on Multimedia*, volume 10, no. 5, 2008: pp. 846–857, doi:10.1109/TMM.2008.922870. Available from: `https://ieeexplore.ieee.org/document/4540196/`

[5] Xue, H.; Li, H. F.; et al. Computationally efficient audio segmentation through a multi-stage BIC approach. In *2010 3rd International Congress on Image and Signal Processing*, volume 8, 2010, pp. 3774–3777, doi:10.1109/CISP.2010.5646687. Available from: `https://ieeexplore.ieee.org/document/5646687/`

[6] Ajmera, J. *Robust Audio Segmentation.* Master's thesis, Indian Institute of Technology Bombay, Lausanne, Switzerland, École Polytechnique Fédérale de Lausanne, 2004. Available from: `https://infoscience.epfl.ch/record/83092/files/rr04-35.pdf`

[7] Kaiser, F. *Music Structure Segmentation.* Dissertation thesis, Technische Universität Berlin, 2012. Available from: https://depositonce.tu-berlin.de/bitstream/11303/3661/1/Dokument_31.pdf

[8] Verma, P.; P., V. T.; et al. Structural Segmentation of Hindustani Concert Audio with Posterior Features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 136–140, doi:10.1109/ICASSP.2015.7177947. Available from: http://compmusic.upf.edu/publications

[9] Theodorou, T.; Mporas, I.; et al. An Overview of Automatic Audio Segmentation. *International Journal of Information Technology and Computer Science*, volume 6, 2014: pp. 1–9, doi:10.5815/ijitcs.2014.11.01. Available from: http://www.mecs-press.org/ijitcs/ijitcs-v6-n11/v6n11-1.html

[10] Sarala, P.; Ishwar, V.; et al. Applause identification and its relevance to archival of Carnatic music. In *Proceedings of the 2nd CompMusic Workshop*, 2012, ISBN 978-84-695-4958-2, pp. 66–71. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.650.4983

[11] Chen, N.; Shijun, W. High-Level Music Descriptor Extraction Algorithm Based on Combination of Multi-Channel CNNs and LSTM. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, ISBN 978-981-11-5179-8, pp. 509–514. Available from: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/17_Paper.pdf

[12] Dieleman, S.; Schrauwen, B. Multiscale Approaches To Music Audio Feature Learning. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013, ISBN 978-0-615-90065-0, pp. 3–8. Available from: http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/69_Paper.pdf

[13] Elizalde, B.; Friedland, G. Lost in segmentation: Three approaches for speech/non-speech detection in consumer-produced videos. In *2013 IEEE International Conference on Multimedia and Expo*, 2013, ISSN 1945-7871, pp. 1–6, doi:10.1109/ICME.2013.6607486. Available from: https://www.icsi.berkeley.edu/pubs/speech/losttranslation13.pdf

[14] Wang, J.-c.; Yen, M.-c.; et al. Automatic Set List Identification and Song Segmentation for Full-Length Concert Videos. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 239–244. Available from: http://www.terasoft.com.tw/conf/ismir2014/

[15] Ryan, Ø.; Dahl, G.; et al. Fourier theory, wavelet analysis and nonlinear optimization. 2012, lecture notes for the course MAT-INF2360 at University of Oslo. Available from: `https://www.uio.no/studier/emner/matnat/math/MAT-INF2360/v12/intro.pdf`

[16] Rao, P. *Audio Signal Processing*, volume 83, chapter 8. Springer-Verlag Berlin Heidelberg, 2007, pp. 169–189, doi:10.1007/978-3-540-75398-8. Available from: `https://www.springer.com/us/book/9783540753971`

[17] Muller, M.; Ellis, D. P. W.; et al. Signal Processing for Music Analysis. *IEEE Journal of Selected Topics in Signal Processing*, volume 5, no. 6, 2011: pp. 1088–1110, doi:10.1109/JSTSP.2011.2112333. Available from: `https://ieeexplore.ieee.org/document/5709966/`

[18] Morin, D. Fourier Analysis, 2009, a chapter of a potential (at the time of writing this thesis) book on Waves. Available from: `http://www.people.fas.harvard.edu/~djmorin/waves/Fourier.pdf`

[19] Brown, J. C. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, volume 89, no. 1, 1991: pp. 425–434, doi:10.1121/1.400476. Available from: `http://academics.wellesley.edu/Physics/brown/pubs/cq1stPaper.pdf`

[20] Varshney, L. R.; Sun, J. Z. Why do we perceive logarithmically? *Significance*, volume 10, 2013: pp. 28–31, doi:10.1111/j.1740-9713.2013.00636.x. Available from: `http://www.rle.mit.edu/stir/documents/VarshneyS_Significance2013.pdf`

[21] Lemmetty, S. *Review of Speech Synthesis Technology.* Master's thesis, Helsinki University of Technology, Espoo, Finland, 1999. Available from: `http://research.spa.aalto.fi/publications/theses/lemmetty_mst/`

[22] Scheirer, E.; Slaney, M. Construction and evaluation of a robust multifeature speech/music discriminator. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, 1997, pp. 1331–1334, doi:10.1109/ICASSP.1997.596192. Available from: `https://ieeexplore.ieee.org/document/596192/`

[23] Dubnov, S. Generalization of spectral flatness measure for non-Gaussian linear processes. *IEEE Signal Processing Letters*, volume 11, no. 8, 2004: pp. 698–701, ISSN 1070-9908, doi:10.1109/LSP.2004.831663. Available from: `https://ieeexplore.ieee.org/document/1316889/`

[24] Abebe, A.; Daniels, J.; et al. Statistics and Data Analysis. 2001, textbook for the course Stat 160 at Wstern Michigan University. Available from: `http://www.stat.wmich.edu/s160/hcopy/book.pdf`

[25] Hyndman, R. J. Moving Averages. *International Encyclopedia of Statistical Science*, 2010: pp. 866–869. Available from: https://robjhyndman.com/papers/movingaverage.pdf

[26] Patro, S. G. K.; Sahu, K. K. Normalization: A Preprocessing Stage. *Computing Research Repository*, 2015. Available from: http://arxiv.org/abs/1503.06462

[27] Wang, D.; Vogt, R.; et al. Automatic audio segmentation using the Generalized Likelihood Ratio. In *Signal Processing and Communication Systems*, 2008, pp. 341–345, doi:10.1109/ICSPCS.2008.4813705. Available from: https://ieeexplore.ieee.org/document/4813705/

[28] Böck, S.; Krebs, F.; et al. Evaluating the Online Capabilities of Onset Detection Methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, ISBN 978-972-752-144-9, pp. 49–54. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.294.3516

[29] Sahidullah, M.; Saha, G. Comparison of Speech Activity Detection Techniques for Speaker Recognition. *Computing Research Repository*, 2012. Available from: http://arxiv.org/abs/1210.0297

[30] Aggarwal, C. C. *An Introduction to Outlier Analysis*, chapter 1. New York, NY: Springer New York, 2013, ISBN 978-1-4614-6396-2, pp. 1–40, doi:10.1007/978-1-4614-6396-2_1. Available from: https://doi.org/10.1007/978-1-4614-6396-2_1

[31] Chandola, V.; Banerjee, A.; et al. Anomaly Detection: A Survey. *ACM Computing Surveys*, volume 41, no. 3, 2009: pp. 15:1–15:58, ISSN 0360-0300, doi:10.1145/1541880.1541882. Available from: http://doi.acm.org/10.1145/1541880.1541882

[32] Re, M.; Valentini, G. *Ensemble methods: A review*, chapter 26. Chapman and Hall/CRC, 01 2012, ISBN 9781138199309, pp. 563–594. Available from: https://www.researchgate.net/publication/230867318_Ensemble_methods_A_review

[33] Müller, M.; Ewert, S.; et al. Making Chroma Features More Robust to Timbre Changes. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, ISBN 978-1-4244-2354-5, ISSN 1520-6149, pp. 1869–1872. Available from: http://resources.mpi-inf.mpg.de/MIR/chromatoolbox/2009_MuellerEwertKreuzer_ChromaFeaturesRobust_ICASSP.pdf

[34] Akosa, J. S. Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data. In *Proceedings of the SAS®Global Forum 2017*

Conference, 2017. Available from: http://support.sas.com/resources/papers/proceedings17/0942-2017.pdf

[35] Peiszer, E.; Lidy, T.; et al. Automatic Audio Segmentation: Segment Boundary and Structure Detection in Popular Music. In *Proceedings of the Second International Workshop on Learning Semantics of Audio Signals*, 07 2008, pp. 45–59, doi:10.1.1.141.2758. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.141.2758

[36] Castán, D.; Tavarez, D.; et al. Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains. *EURASIP Journal on Audio, Speech, and Music Processing*, volume 2015, no. 1, 2015: pp. 33–41, ISSN 1687-4722, doi:10.1186/s13636-015-0076-3. Available from: https://doi.org/10.1186/s13636-015-0076-3

[37] Senin, P. Dynamic Time Warping Algorithm Review. Technical report, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, 2008. Available from: http://csdl.ics.hawaii.edu/techreports/2008/08-04/08-04.pdf

[38] Jones, E.; Oliphant, T.; et al. SciPy: Open source scientific tools for Python. 2001–. Available from: http://www.scipy.org/

[39] McFee, B.; McVicar, M.; et al. librosa/librosa: 0.6.0. Feb. 2018, doi:10.5281/zenodo.1174893. Available from: https://doi.org/10.5281/zenodo.1174893

[40] Pedregosa, F.; Varoquaux, G.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, volume 12, 2011: pp. 2825–2830.

Appendix A

Acronyms

BIC Bayesian Information Criterion

CQT Constant-Q Transform

DFT Discrete Fourier Transform

DTW Dynamic Time Warping

FFT Fast Fourier Transform

GLR Generalized Likelihood Ratio

GMM Gaussian Mixture Model

HMM Hidden Markov Model

LLR Log-Likelihood Ratio

MFCC Mel-Frequency Cepstral Coefficients

MIR Music Information Retrieval

MIREX Music Information Retrieval Evaluation Exchange

NDTWE Normalized Dynamic Time Warping Error

RMS Root-Mean-Square Energy

SER Segmentation Error Score

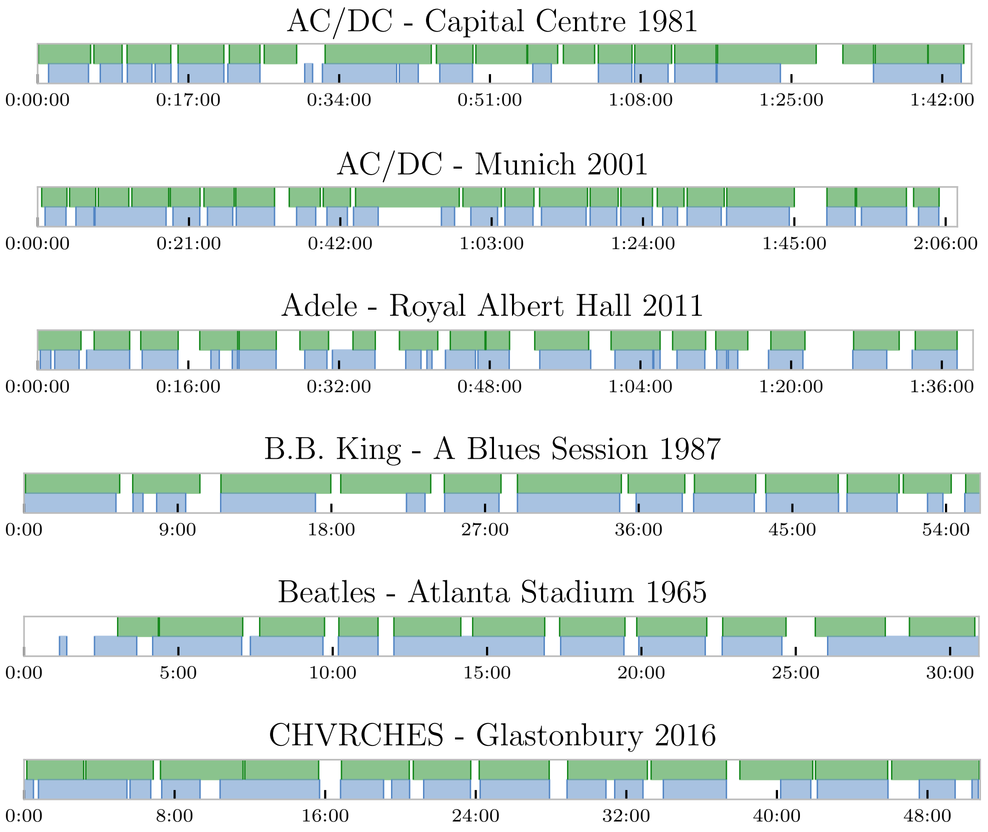
STFT Short-Time Fourier Transform

SVM Support Vector Machine

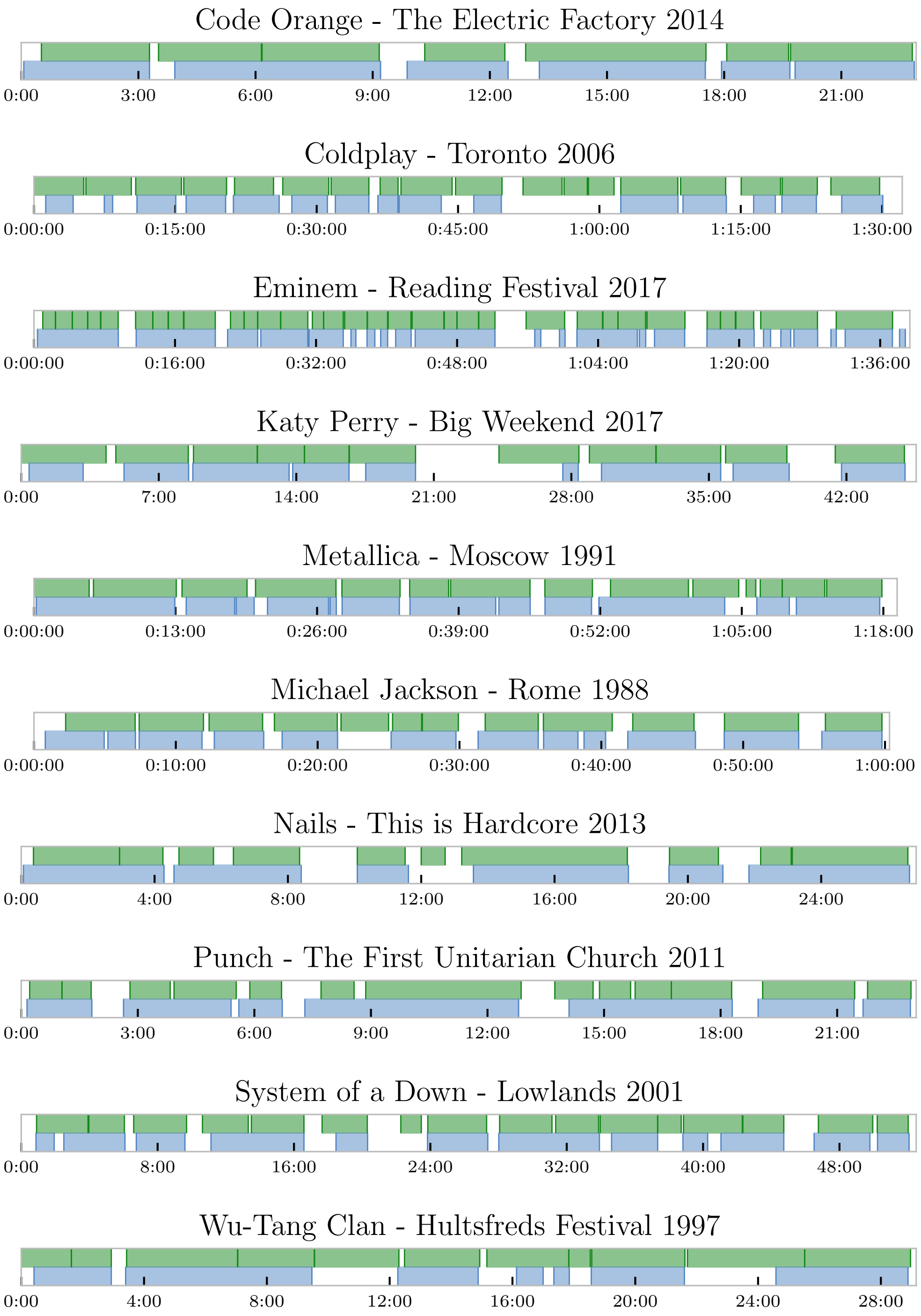
ZCR Zero Crossing Rate

Appendix **B**

Results Visualization



Green segments represent the ground truth, blue segments correspond to the estimatation.



Green segments represent the ground truth, blue segments correspond to the estimatation.

Appendix C

Contents of Enclosed CD

```
 readme.txt...the file with the thesis overview and execution instructions
 src......................................the directory of source codes
   impl......................the directory with implementation sources
   thesis..............the directory of LaTeX source codes of the thesis
 thesis.pdf............................the thesis text in PDF format
```