Music Emotion Classification

Nikolaos Nikolaou

A dissertation for the Diploma of Electronic & Computer Engineer

Department of Electronic & Computer Engineering
Technical University of Crete

Examination committee:

Associate Prof. Alexandros Potamianos (Supervisor)
Prof. Vassilios V. Digalakis
Assistant Prof. Michail G. Lagoudakis

Chania 2011
Crete, Greece

ABSTRACT

In this thesis we focus on the automatic emotion classification of music samples. We extract a set of features from the music signal and examine their discriminatory capability using various classification techniques. Our goal is to determine the features and the classification methods that lead to the best classification of the emotion a music sample conveys. During the course of the thesis, we generated our own dataset of annotated song samples and we examined two distinct methods of describing an emotion: using clusters consisting of various emotional states, and using a two-dimensional representation of the emotion in the Valence-Activation plane. The latter method was chosen as the most successful. We also tried other approaches of music emotion classification (MEC) as well, such as treating the song sample as an amplitude and frequency modulated (AM-FM) signal, on which we subsequently perform multiband demodulation analysis (MDA) testing various Gabor filter banks (Mel scale-based filter bank , Bark scale-based filter bank, and a number of fractional octave-based filter banks). Statistics of the Frequency Modulation Percentages (FMPs) of each band derived from the demodulation, proved to be quite successful features in the classification of emotion. Finally, we explored other modalities besides the music sound signal itself, such as a number of features derived from the chords of the song samples, classification of the song samples' lyrics using various techniques and a brief investigation of Electroencephalogram (EEG) data generated by one of the annotators while performing the annotation of the song samples. Our final feature-pack included a combination of the most successful features among the ones we studied: (i) music-inspired features (features based on music theory and psychoacoustics, derived from either the sound signal or the chords of the sample), (ii) statistics of the FMPs and (iii) statistics of the Mel-frequency cepstral coefficients (MFCCs). This feature-pack proved to be more robust than its three individual components and in the end we achieved results that reached 85.7% correct classification rate in the dimension of Valence and 85.1% correct classification rate in the dimension of Activation. We finally demonstrate that by discarding training samples that are assigned a label too close to the neutral value, our results can improve even further, especially in the dimension of Activation.

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή, επικεντρωνόμαστε στην αυτόματη ταξινόμηση μουσικών αποσπασμάτων με βάση το συναίσθημα. Εξάγουμε ένα σύνολο χαρακτηριστικών από το μουσικό σήμα και ερευνούμε την ικανότητά τους να διαχωρίσουν τις κατηγορίες του συναισθήματος, χρησιμοποιώντας διάφορες τεχνικές ταξινόμησης. Στόχος μας είναι να εξακριβώσουμε τα χαρακτηριστικά και τις μεθόδους εκείνες που οδηγούν στην καλύτερη ταξινόμηση του συναισθήματος που μεταδίδει ένα μουσικό απόσπασμα. Κατά τη διάρκεια της εργασίας, δημιουργήσαμε το δικό μας σύνολο από προσημειωμένα μουσικά αποσπάσματα και μελετήσαμε δύο ξεχωριστές μεθόδους περιγραφής των συναισθημάτων: χρήση κατηγοριών αποτελούμενων από διάφορα συναισθήματα, και χρήση μιας δισδιάστατης αναπαράστασης του συναισθήματος στους άξονες της Χαράς/Λύπης (Valence) και Ενεργοποίησης/Απενεργοποίησης (Activation). Η τελευταία μέθοδος επιλέχθηκε τελικά ως η πιο επιτυχής. Επιπλέον, δοκιμάσαμε και άλλες προσεγγίσεις της ταξινόμησης μουσικών αποσπασμάτων με βάση το συναίσθημα, όπως να αντιμετωπίσουμε το μουσικό απόσπασμα ως ένα σήμα διαμορφωμένο στο πλάτος και τη συχνότητα (amplitude and frequency modulated (AM-FM) signal), στο οποίο στη συνέχεια εφαρμόζουμε multiband demodulation analysis (MDA) δοκιμάζοντας διάφορες συστοιχίες φίλτρων Gabor (βασισμένη στην κλίμακα Mel, βασισμένη στην κλίμακα Bark, και άλλες, βασισμένες σε fractional octave φίλτρα). Τα στατιστικά μεγέθη των Frequency Modulation Percentages (FMPs) κάθε ζώνης συχνοτήτων που προέκυψαν από την αποδιαμόρφωση, αποδείχθηκαν αρκετά επιτυχημένα χαρακτηριστικά στην ταξινόμηση του συναισθήματος. Τέλος, εξερευνήσαμε και άλλες τροπικότητες (modalities) πέρα από το μουσικό σήμα ήχου αυτό καθεαυτό, όπως ένα πλήθος χαρακτηριστικών που προήλθαν από τις συγχορδίες των αποσπασμάτων, ταξινόμηση των στίχων τους με τη χρήση διάφορων τεχνικών, και μια σύντομη διερεύνηση δεδομένων του Ηλεκτροεγκεφαλογραφήματος (EEG) ενός από τους προσημειωτές κατά τη διάρκεια της προσημείωσης των αποσπασμάτων. Το τελικό μας πακέτο χαρακτηριστικών περιελάμβανε ένα συνδυασμό των πιο επιτυχημένων χαρακτηριστικών μεταξύ όσων μελετήσαμε: (i) χαρακτηριστικά εμπνευσμένα από τη μουσική (με βάσεις στη μουσική θεωρία και την ψυχοακουστική, προερχόμενα είτε από το σήμα ήχου είτε από τις συγχορδίες), (ii) στατιστικά μεγέθη των FMPs και (iii) στατιστικά μεγέθη των Mel-frequency cepstral coefficients (MFCCs). Αυτό το πακέτο χαρακτηριστικών αποδείχθηκε πιο στιβαρό από τις τρεις επιμέρους συνιστώσες του και στο τέλος πετύχαμε έως και 85.7% ποσοστό επιτυχούς ταξινόμησης στη διάσταση Χαράς/Λύπης και 85.1% ποσοστό επιτυχούς ταξινόμησης στη διάσταση Ενεργοποίησης/Απενεργοποίησης. Τέλος δείξαμε ότι εξαιρώντας δείγματα εκπαίδευσης με ετικέτες πολύ κοντά στην ουδέτερη τιμή, τα αποτελέσματά μας βελτιώνονται ακόμα περισσότερο, ιδιαίτερα στη διάσταση Ενεργοποίησης/Απενεργοποίησης.

ACKNOWLEDGMENTS

I thank my family and all my friends for their love and support. I would also like to thank my supervisor Prof. Alexandros Potamianos for his guidance and support. Finally, I am grateful to all those who helped me throughout the various steps of this thesis, especially the people at the Information Processing and Computer Networks Laboratory of the Technical University of Crete (special thanks to Nikos Malandrakis for his invaluable help throughout my work) and all those who volunteered for the annotation process.

TABLE OF CONTENTS

1.INTRODUCTION

In this section we will discuss the motivation behind studying the effect of music on emotion and striving for more successful systems that perform automatic music emotion classification, as well as present some possible practical applications of such systems in everyday life -among other fields. We will then move on to a brief overview of the present thesis, analyzing its basic stages and highlighting the individual chapters.

1.1 Motivation and Possible Applications

Music Information Retrieval is a sub-area of information retrieval (IR), the area of study concerned with searching for documents, for information within documents, and for metadata about documents, as well as that of searching relational databases and the World Wide Web. Music, still remains a relatively unexplored area for the application of IR techniques and therefore offers promising ground for research and exploration. Important research directions include for example similarity retrieval, musical genre classification, or music analysis and knowledge representation [1].

Automatic Emotion Classification and Automatic Emotion Recognition are also becoming areas of increasing research activity, as the importance of emotion becomes more and more widely recognised. Emotion, undoubtably plays an important part in our everyday life. Our emotional state affects everything, from our performance in our tasks, to the very way we preceive the world and our respond to its stimuli. However, emotion being a result of the inner workings of our brain, has yet to be fully understood and poses a challenge to psychologists and computer engineers alike. The former group strives towards a better understanding of the emotions in itself, while the latter focuses on making use of emotion-related information mainly in the context of Human Computer Interfaces (HCI) and in IR tasks.

Music Emotion Classification (MEC) lies on the intersection of these two very interesting and challenging domains of research. As the number of music recordings increases exponentially, automatic classification of music emotion becomes one of the hot spots on research and engineering [1]. This kind of categorization is particularly important because of its direct link with one of the primary motives to listen to music: to feel emotions [3].

Although automatic music mood classification seems a difficult task at first glance, recent studies have shown that it is possible to a certain extent. In fact, a MEC system can even give satisfying results if we consider a few simple categories, if we check for valid agreements between people [2] and if we concentrate on meaningful features.

In this thesis we focus on the feature selection aspect of such a system. Our goal is to determine the features that lead to the best classification of the emotion a music sample[1] conveys. Using these features a system that performs MEC with a high accuracy can be implemented.

The results of such a study can be useful in their own merit. They can help us better understand music, emotion and the link between them. For musicians, our findings can prove useful as a quantitative indicator as to what constitutes a 'happy' or an 'arousing' song, complementing their own experiences

1.Statisticians, always refer to a 'sample' as a collection of items, and discuss 'a sample of size n'. We, however, will use the term 'sample' to refer to a single song excerpt (a particular item of our collection), unless explicitly stated otherwise.

and helping them add elements that allow their compositions to better convey a desired emotion. For psychologists the results are another window to the human psyche, providing them with insights about the world of emotions.

However, an engineer would be far more interested in everyday practical implementations of a MEC system. Such a system could be used for instance in an emotion based music retrieval context. For example, a system that could connect to a database, or the World Wide Web and search for songs of a particular emotional profile, perhaps trained on the preferences of its user. Such systems could also be trained to make suggestions of songs or entire playlists that have an emotional profile similar to the one implied by the user's current favourite songs or explicitly demanded by the user (e.g. "Suggest me a 'happy' list of songs").

Another similar application could be creating a program that organises existing databases of songs into emotional categories based on its estimation about the songs' emotional profile, making the task of managing such adatabase easier.

Automatic song composition programs could be produced using the best features we discuss here as a guideline to steer towards a composition that successfully conveys a certain emotion.

Perhaps some of the features we study could also find successful application in speech classification or recognition tasks, despite the fact that most of them are musically-inspired. Speech and Music are both perceived by hearing, and hearing evolved hand in hand with speech.

Last but not least, another domain where a robust set of music emotion classification features can find applications is in similar emotion classification tasks in movies. Movies combine sound among other modalities (visual, perhaps textual through their subtitles). To correctly perform movie emotion classification, we need to incorporate a number of meaningful features from all these modalities and the auditory information, besides the actors speech and the various auditory effects also contains music as an itegral part of it.

1.2 Outline of the Diploma Thesis

The rest of the chapters are organised as follows:

In **Chapter 2** we cover the basics of the theoretical background of the thesis. We first break down Music Emotion Classification to its individual components: Music, Emotion and Classification. In each of these subsections we define pieces of the fundamental framework in which we will work. However, we are not going to delve into too many details, as all three components constitute entire fields of study in their own right. In the first subsection we try to understand what music is, how it is generated, how the human perception of music (auditory system and psychoacoustics) works. We then move on to introducing the fundamental mathematics of a sound signal and the mathematics of music in particular, music theory. In the 'Emotion' sub-chapter, we present the reader with some of the greatest challenges of emotion categorisation and we also analyse the techniques we used in order to represent emotion in our work (categorisation using clusters consisting of various emotional states, and categorisation using a two-dimensional representation of the emotion in the Valence-Activation plane). Finally, in the 'Classification' subsection, we present the definition of a classification problem and the theory behind the particular classifiers we used in the classification stage.

Chapter 3 focuses on the datasets used or created during the course of this thesis, as well as the programming tools and environments utilised. We present the format of the data, the pre-processing

procedure the initial data underwent and the steps towards extracting information in the form of chord, lyrics, and EEG data. Finally, we also offer a brief overview of the theory behind the EEG and the various brain rythms (brain waves).

In **Chapter 4** the annotation process is presented, as well as the labelling schemes used : MIREX mood clusters, Self Assessment Manequinns (SAMs). We will also discuss the agreement evaluation techniques and the methods of assigning the final labels (the ones to be used in the classification stage) to each song for each labeling scheme. Annotators' personal, pairwise and overall statistics are also presented in an effort to better understand the underlying peculiarities of our problem. Which emotion do the annotators favor? What is the overall agreement amongst all annotators? What is the form of the confusion matrix (for mood clusters)? Do the means of the labels in valence and activation (for the two-dimensional emotion resentation) fall near the median or not? All these questions and many more are answered in this chapter. Finally we present the reasons that made us decide to continue in the next steps with only the labels produced by the SAMs labeling scheme, therefore consindering from now on only the two-dimensional (Valence axis versus Arousal axis) emotion resentation.

Chapter 5 is dedicated to the presentation of the various features we experimented with during our work. The chapter is divided into subsections according to the modality from which the feature was extracted. Sound signal features**,** chord features, EEG features and lyrics classification each have a susection of this chapter dedicated to them. The features derived from the AM-FM model of the sound signal, the Frequency Modulation Percentages (FMPs) and their statistics were deemed worthy of an entire chapter for themselves, since they represent a different approach to the modeling of the sound signal and the theory of AM-FM signals deserved some further analysis.

In **Chapter 6** we discuss the outline of the classification process. We present the attribute-relationship file format (ARFF) and we will explain how the feature selection and the model training and model evaluation stages are carried out.

Chapter 7 contains the results of the classification, along with some interesting comments and observations, as well as an analysis of how the classification results improve as we use more and more 'certain' samples of our sample space.

Finally, in **Chapter 8** we discuss our conclusions, the strengths and weaknesses of our approach and we suggest some of the possible directions towards improving upon this work, or furthering its findings and applications.

1.3 Contributions

During the course of this thesis, we created two annotated (on the emotional dimensions of Valence and Activation) musical datasets. The first one consists of 181 classical music excerpts of 20 seconds duration, the second one of 412 samples of songs by The Beatles of 10-20 seconds duration each, complemented by their corresponding chords and lyrics in separate files. Furthermore, we developed the tools to add more annotators if we wish so.

We explored a variety of features whose basis lies in music theory and psychoacoustics and which are used extensively in music emotion classification, as well as features mostly used in speech recognition/classification tasks, such as MFCCs and FMPs, and of course the combination of all these.

The music-inspired features were extracted from the sound signal using MIRtoolbox [37]. To them we added a set of features extracted directly from the samples' chords, which proved to complement them

very well leading to more successful classification results.

For the calculation of the FMPs we were based on previous work by P. Maragos [69],[70],[71], A.Potamianos [70], [71], D. V. Dimitriadis [71],[80], and P. Tsiakoulis [80]. We adapted it to music emotion classification, using a number of music (fractional octave) and hearing (bark-scale based) grounded filterbanks to perform the multiband demodulation analysis step.

The combination of these feature-packs, provided us with better results than any of the individual components, and as we will see in **chapter 7.3**, they appear to lead to more robust classification, especially in the dimension of Activation.

We also experimented with other modalities, such as lyrics classification (with the aid of N. Malandrakis [79]) and EEG features classification with less successful results. Still these modalities are worthy of further study.

2.THEORETICAL BACKGROUND

2.1 Music

What is music? The question might seem trivial at first, but the more we think about it, the harder it gets to answer it. Suffice it to mention that the definition of music has been tackled by philosophers of art, lexicographers, composers, music critics, musicians, semiologists, linguists, sociologists, and neurologists and still evades a proper consensus.

The reason is that the perception of music changes from culture to culture both through time and through space. Thus we realize that it has cultural and sociological implications. It slso varies from person to person or even changes as a person grows older, so psychological, or if we delve even deeper biological and neurological implications come into play as well. Music is a form of art, but undoubtably it is also very mathematical, as we will see in the rest of this chapter. Music is a communicative activity which conveys to the listener moods, emotions, thoughts, impressions, or philosophical, sexual, or political concepts or positions.

As any other thing we, as biological organisms, can and tend to do, preceiving sound in general, as well as in particular preceiving certain sounds, combinations or sequences of sounds as nice and reassuring, while others cause us discomfort and sadness, can also be examined in the context of biological evolution.

Music, at its core, is a byproduct of our evolution. Evolving sound perceptors helped our early ancestors perceive and hunt their food and perceive and avoid predators. After they evolved into sexually-reproducing organisms, sound perception and production must have also come in handy while searching for a possible mate. As they developed into more and more complex organisms they started developing parental skills. Now hearing the distress calls of their young allowed them to know if they were hungry or in danger and rush to their aid. Finally, our ancestors evolved to become social animals, they banded together in groups and evolved speech, a tool that helped them better organise as a group, avoid internal conflict and develop deeper social relationships. At one point music started having a life of its own and singing and playing music became a form of art and communication.

So it is no surprise that even the simplest, most commonly accepted definitions of what music is fail to convey its entirety. One could argue that music is the science and art of temporally organized sound and silence. Others would prefer the following definition: "Music is a sequence of sounds with a particular rythm.". Others, yet, would find both definitions inaproppriate...

However, no one can deny that sound is the 'alphabet' of music and its medium of transmition. So let us now study the fundamental physics and mathematics of sound...

2.1.1 Production of sound

A very good description of the physics of sound can be found at Dave Benson's Book 'Music: A Mathematical Offering' [4]. Here we will cover only the basic principles behind sound production and transmission based upon it.

<u>A disturbance in the air</u>
Sound consists of vibrations of the air. Air is a gas, which means its atoms and molecules are not in

such close proximity to each other as they are in a solid or a liquid. The molecules at room temperature under normal conditions travel with a mean velocity of around 450–500 meters per second. The mean free path of an air molecule is 6×10^{-8} meters. This means that on average, an air molecule travels this distance before colliding with another air molecule. The collisions between air molecules are perfectly elastic, so this does not slow them down.

The collision frequency of a given air molecule is given by:

$$collision\ frequency = \frac{mean\ velocity}{nean\ free\ path} \approx 10^{10}\ collisions\ per\ second$$

That is the reason why air molecules don't just fall down on the ground: they don't get very far down before being bounced back up again. The effect of gravity is observable just as a gradation of air pressure, so that if we go up to a high elevation, the air pressure is noticeably lower.

When an object vibrates, it causes waves of increased and decreased pressure in the air. These waves are perceived by the ear as sound, in a manner we will discuss in the next part of this chapter.
Sound travels through the air at about 340 meters per second. This does not mean that any particular molecule of air is moving in the direction of the wave at this speed, but rather that the local disturbance to the pressure propagates at this speed.

Waves in nature are divided into different categories. In some waves, for example water waves, the local movements involved in the wave are up and down, which is at right angles to the direction of propagation of the wave. Such waves are called transverse waves (**fig. 2.1a**). Electromagnetic waves are also transverse. In the case of sound, on the other hand, the motions involved in the wave are in the same direction as the propagation. Waves with this property are called longitudinal waves (**fig. 2.1b**).
Another way to categorize waves is on the basis of their ability or inability to transmit energy through a vacuum (empty space). Categorizing waves on this basis leads to two notable categories: electromagnetic waves and mechanical waves. Mechanical waves need a medium through which to propagate. Sound, for instance is such a wave and air such a medium. Electromagnetic waves, such as visible light, on the other hand, can travel through vacuum -in fact they do so very fast.

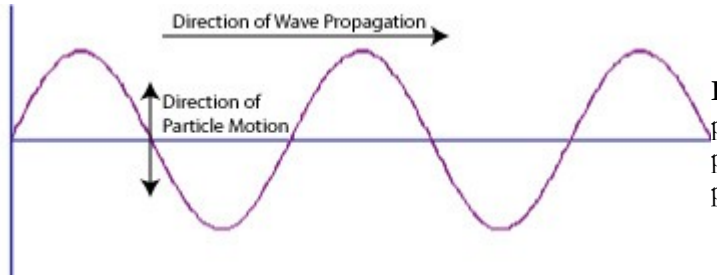


Figure 2.1a:. A transverse wave. The particles move in a direction that is perpendicular to the direction of wave propagation.

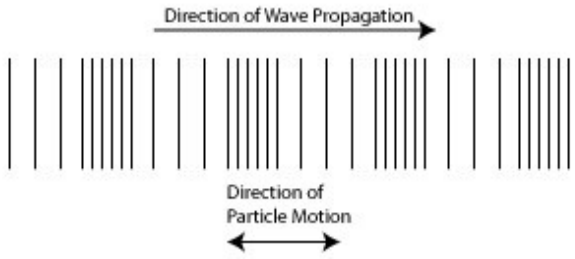


Figure 2.1b: A longitudinal wave, made up of compressions - areas where particles are close together - and rarefactions - areas where particles are spread out. The particles move in a direction that is parallel to the direction of wave propagation. Sound waves belong to this category.

Figure 2.1: Transverse and longitudinal waves.
(Images adapted from: http://www.visionlearning.com/library/module_viewer.php?mid=102)

Sound waves have four main attributes which affect the way they are perceived (table 2.1). The first is amplitude, which means the size of the vibration, and is perceived as loudness. The amplitude of a typical everyday sound is very small, usually only a mere fraction of a millimeter. The second attribute is pitch, which should at first be thought of as corresponding to the frequency of vibration. The third is timbre, which corresponds to the shape of the frequency spectrum of the sound. The fourth is duration, which means the length of time for which the note sounds [4].

These notions need to be modified for a number of reasons. The first is that most vibrations do not consist of a single frequency, and naming a 'defining' frequency can be difficult. The second related issue is that these attributes should really be defined in terms of the perception of the sound, and not in terms of the sound itself. So for example the perceived pitch of a sound can represent a frequency not actually present in the waveform. This phenomenon is called the 'missing fundamental', and is part of the subject of psychoacoustics.

Attributes of sound

Physical	Perceptual
Amplitude	Loudness
Frequency	Pitch
Spectrum	Timbre
Duration	Length

Table 2.1: A table showing the physical attributes of the sound wave and the perceptual attributes to which they roughly correspond. (Table taken from Benson, D.: *Music: A Mathematical Offering* Cambridge University Press, 3rd Printing (2008))

Harmonic motion

Until now we covered the basic principles behind sound transmission. It is merely a distortion of air molecules that travels in space. But how is this disturbance created in the first place and how does our auditory system convert compressions and rarefactions of air molecules into something as meaningful as music or speech?

To answer both of these questions let us begin with remembering the basic mathematical properties of harmonic motion, as both the production of sound and the human sound perceptors are governed by various types of oscillations. On the one hand we have the oscillations that generate the sound (e.g. vibrations of the vocal folds, various types of vibrations caused by musical instruments) . On the other, we have the vibrations inside the human auditory system that 'decode' the sound signal into neural activity thus causing us to 'understand' sound as we know it. Again, Benson [4] explains this aspect of sound very comprehensively, and we present here the basic theory without delving into too much detail.

When a particle of mass m is subject to a force F towards the equilibrium position, $y=0$, whose magnitude is proportional to the distance y from the equilibrium position,

$$F=-ky$$

Where, k is just the constant of proportionality. Newton's laws of motion give us the equation

$$F=ma$$

Where:

$$a=\frac{d^2y}{dt^2}$$

is the acceleration of the particle and **t** represents time.

Combining these equations, we obtain the second order differential equation

$$\frac{d^2y}{dt^2}+\frac{ky}{m}=0 \quad \textbf{(Eq. 2.1.1a)}$$

Or equivalently:

$$\ddot{y}+\frac{ky}{m}=0 \quad \textbf{(Eq. 2.1.1b)}$$

using the dot notation for the first and second order derivatives.

The solutions to this equation are the functions of the form:

$$y=A\cos(\sqrt{\frac{k}{m}}t)+B\sin(\sqrt{\frac{k}{m}}t) \qquad \textbf{(Eq. 2.1.2)}$$

The fact that these are the solutions of this differential equation is the explanation of why the sine wave, and not some other periodically oscillating wave, is the basis for harmonic analysis of periodic waves. For this is the differential equation approximating the movement of any particular point on the basilar membrane in the cochlea, and hence governing the human perception of sound. We say 'approximating', because a number of over-simplifications were made [4]: (i) **Eq. 2.1.1** is an ordinary differential equation. In reality, a second order partial differential equation describes the motion of the surface of the basilar membrane. This does not really affect the results of the analysis much except to explain the origins of the constant *k*. (ii) The motion is really not a simple harmonic motion, but rather a forced damped harmonic motion in which there is a damping term proportional to velocity, coming from the viscosity of the fluid and the fact that the basilar membrane is not perfectly elastic. (iii) For loud enough sounds the restoring force may be nonlinear, this is not taken into account in **Eq. 2.1.1**. (iv) Most musical notes do not consist of a single sine wave. For example, if a string is plucked, a periodic wave will result, but it will usually consist of a sum of sine waves with various amplitudes. So there will be various different peaks of amplitude of vibration of the basilar membrane, and a more complex signal is sent to the brain.

What happens when a string vibrates

As for the generation of sound in the first place, vibrations obeying the laws of harmonic motion are also responsible for it as well, as we already hinted above, though generally not as simple as the case we just presented.

Let us consider a vibrating string, anchored at both ends. Let us suppose at first that the string has a heavy bead attached to the middle of it, so that the mass *m* of the bead is much greater than the mass of the string. Then the string exerts a force *F* on the bead towards the equilibrium position whose magnitude, at least for small displacements, is proportional to the distance *y* from the equilibrium position,

$$F=-ky$$

As we already shown, we obtain the differential equation (Eq. 2.1.1) whose solutions are the functions generated by **Eq. 2.1.2**. The constants A and B of **Eq.2.1.2** are determined by the initial position and velocity of the bead.

If the mass of the string is uniformly distributed, then more vibrational 'modes' are possible. In general, a plucked string will vibrate with a mixture of all the modes described by multiples of the natural frequency, with various amplitudes. The amplitudes involved depend on the exact manner in which the string is plucked or struck. For example, a string struck by a hammer, as happens in a piano, will have a different set of amplitudes than that of a plucked string. The general equation of motion of a typical point on the string will be:

$$y=\sum_{n=1}^{\infty}\left(A_n\cos\left(n\sqrt{\frac{k}{m}}t\right)+B_n\sin\left(n\sqrt{\frac{k}{m}}t\right)\right)$$

So a string vibrates with a number of different frequencies at the same time. Decomposing a periodic wave as a sum of sine waves is the subject of the theory of Fourier Series.

Forced harmonic motion

In the real world, however, harmonic motion is rarely as ideal as we discussed above. Damped harmonic motion arises when in addition to the restoring force , $F=-ky$, there is a frictional force proportional to velocity:

$$F=-ky-\mu\dot{y}$$

For positive values of μ, the extra term damps the motion, while for negative values of μ it promotes or forces the harmonic motion. In this case, the differential equation we obtain is

$$m\ddot{y}+\mu\dot{y}+ky=0 \qquad \textbf{(Eq. 2.1.3)}$$

Forced harmonic motion is where there is a forcing term $f(t)$ added into **Eq. 2.1.3** to give an equation of the form

$$m\ddot{y}+\mu\dot{y}+ky=f(t) \quad \textbf{(Eq. 2.1.4)}$$

This represents a damped system with an external stimulus **f(t)** applied to it. We are particularly interested in the case where $f(t)$ is a sine wave, because this represents forced harmonic motion. Forced harmonic motion is responsible for the production of sound in most musical instruments, as well as the perception of sound in the cochlea. Forced harmonic motion is what gives rise to the phenomenon of resonance.

The case of forced harmonic motion of interest to us is the equation

$$m\ddot{y}+\mu\dot{y}+ky=R\cos(\omega t+\varphi) \qquad \textbf{(Eq. 2.1.5)}$$

Eq. 2.1.5 represents a damped harmonic motion with a forcing term of amplitude *R* and angular velocity ω. The term φ corresponds to a phase angle.

The solution of **Eq. 2.1.5** is:

$$y=\frac{R\cdot e^{i(\omega t+\varphi)}}{-m\omega^2+i\mu\omega+k}$$

The amplitude of the resulting vibration, and therefore the degree of resonance (since we started with a forcing term of unit amplitude) is given by taking the absolute value of this solution,

$$|y| = \frac{R}{\sqrt{(k - m\omega^2)^2 + \mu^2 \omega^2}}$$

This amplitude magnification reaches its maximum when

$$\omega = \sqrt{\frac{k}{m} - \frac{\mu^2}{2m^2}} \qquad \textbf{(Eq. 2.1.7)}$$

when we have amplitude

$$\frac{mR}{(\mu\sqrt{km - \frac{\mu^2}{4}})}$$

The value of ω given by **Eq. 2.1.7** is called the resonant frequency of the system. To convert it from angular frequency ω to circular frequency f we just need to divide by 2π:

$$f = \frac{\omega}{2\pi}$$

Vibrations of the musical instruments

The resonant frequencies of the vibrations produced by the various musical instruments determine their unique sounds. These vibrations are modeled by wave equations (in **Eq. 2.1.8** we can see a one-dimensional wave equation) as we need to regard for example the displacement y of a point of a string as a function both of time t and position x along the string.

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} \quad \textbf{(Eq. 2.1.8)}$$

As we can see, now partial derivatives come into play. Furthermore, the dimensions we must take into account for some instruments can be more than one: two or even four [4] .To add even more complexity, even chaos phenomena can arise, even in our own vocal folds [15]. So at this point we will end our discussion about the production of sound as we already strayed a bit more far into the subject than we should have. As a footnote, we present a classification of the musical instruments into five main categories -which correspond reasonably well to the mathematical description of the sound they produce- conducted by [5] and extended by [6] (The latter added the electrophones).

CATEGORY	THE SOUND IS PRODUCED...	EXAMPLES OF INSTRUMENTS
Idiophones	...by the body of a vibrating instrument	Xylophone, Cymbals
Membranophones	...by the vibration of a stretched membrane	Drums
Chordophones	...by one or more vibrating strings.	Violin, Guitar, Piano
Aerophones	...by a vibrating column of air	Flute, Trombone, Vocal Tract
Electrophones	...primarily by electrical or electronic means	Electronic Synthesizer, Computer Programs

Table 2.2: A table showing the five main categories of musical instruments based on the way the sound is produced by them

2.1.2 Human Perception of Sound

Limits of hearing

We already mentioned intensity, frequency, duration and spectrum as the basic physical attributes used to describe the acoustical properties of a sound. These attributes do not form music itself, but they can vary the perception of each sound components of a musical flow. The perceptual attributes, pitch, loudness, and timbre, are the ones that describe how the physical attributes related to sound are perceived and interpreted as a mental construct by the brain, through our auditory system.

Since sound is carried by waves propagating through a medium such as air, the detection of these vibrations constitutes our sense of hearing. The subfield of psychophysics (the study of psychological responses to physical stimuli) studying the relationship between musical stimuli and the induced mental responses is called psychoacoustics and associating the physical attributes of a sound wave and the aforementioned perceptual attributes is one of its main focuses.

Intensity is proportional to energy, i.e. the variance of air pressure, in a sound wave. Sound intensity is measured in terms of Sound Pressure Level (SPL) or Sound Level Lp on a logarithmic scale, thus the result can be expressed in deciBels (dB), when so, it is often also mentioned as :dBSPL

$$L_p = 10\log_{10}\left(\frac{p_{rms}^2}{p_{ref}^2}\right) = 20\log_{10}\left(\frac{p_{rms}}{p_{ref}}\right)$$

where p_{ref} is the reference sound pressure and p_{rms} is the rms[2] sound pressure being measured [5].

Usually i p_{ref} s taken equal to p_o ,the estimated threshold of hearing at 1KHz. The threshold of hearing is generally reported as the rms sound pressure of 20μPa, which is approximately the quietest sound a young human with undamaged hearing can detect at 1KHz. SPL is inversely proportional to distance from the sound sources.

Loudness is the perceptual attribute related to changes in intensity, that is, increases in sound intensity are perceived as increases in the loudness mechanism. Unfortunately this relationship is not a trivial one. Loudness also depends on other factors like spectrum, duration and presence of background sounds.

Winckel[8] proposed the range of hearing for a young adult human ear, shown in (**fig. 2.2**). This range can vary with age and an individual's sensitivity. Winckel's range of hearing is valid for sustained sine tones. For shorter tones this threshold can raise, this is because, approaching to the borders of the threshold, the ear seems to integrate energy for shorter tones, at least for less than 200ms[9]. 20 Hz is considered the normal low frequency limit of human hearing. When pure sine waves are reproduced under ideal conditions and at very high volume, a human listener will be able to identify tones as low as 12 Hz[10]. Below 10 Hz it is possible to perceive the single cycles of the sound, along with a sensation of pressure at the eardrums. In other words, the human body can sense very low frequencies, although ears do not, and that the upper limit of sensitivity may be well beyond 20KHz.

---

2 The root mean square (rms) or quadratic mean, is a statistical measure of the magnitude of a varying quantity. In the case of a set of n values, $\{x_1, x_2, \ldots, x_n\}$, the RMS value is given by: $x_{rms} = \sqrt{\frac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}}$

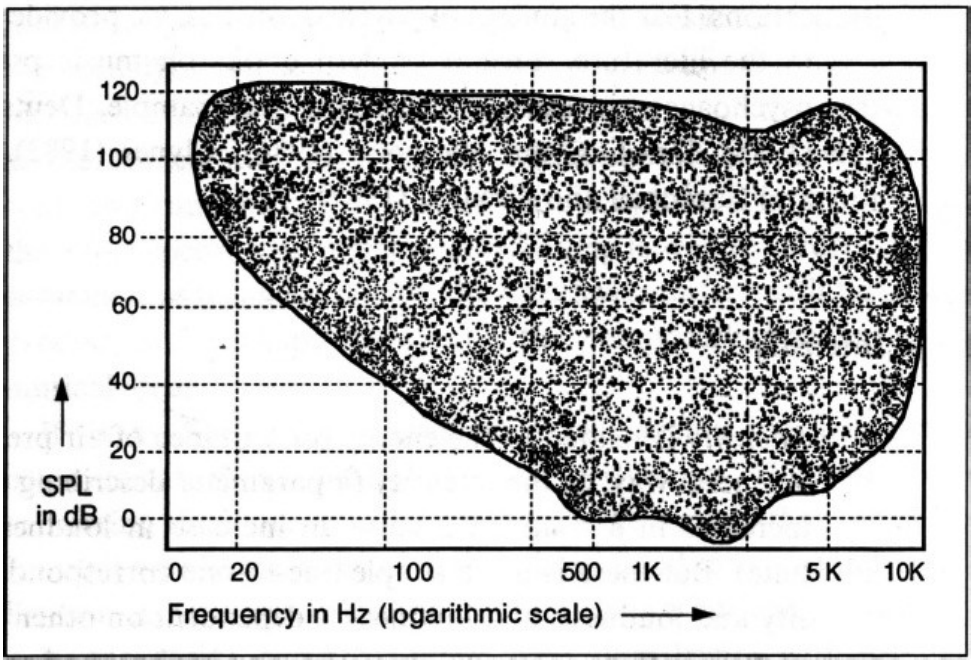


Figure 2.2: Winckel's threshold of hearing. It shows how the Sound Pressure Level (SPL), a measure of the sound intensity is related to the frequency of the sound.(Image from: Winckel, F.: *Music, Sound and Sensation: A Modern Exposition*, Dover (1967))

Another useful tool are the Fletcher-Munson curves (**fig. 2.3**) They proposed a graph similar to that of Winckel, introducing the concept of constant-loudness contours. The meaning of this graph is that each curve has roughly the same loudness.

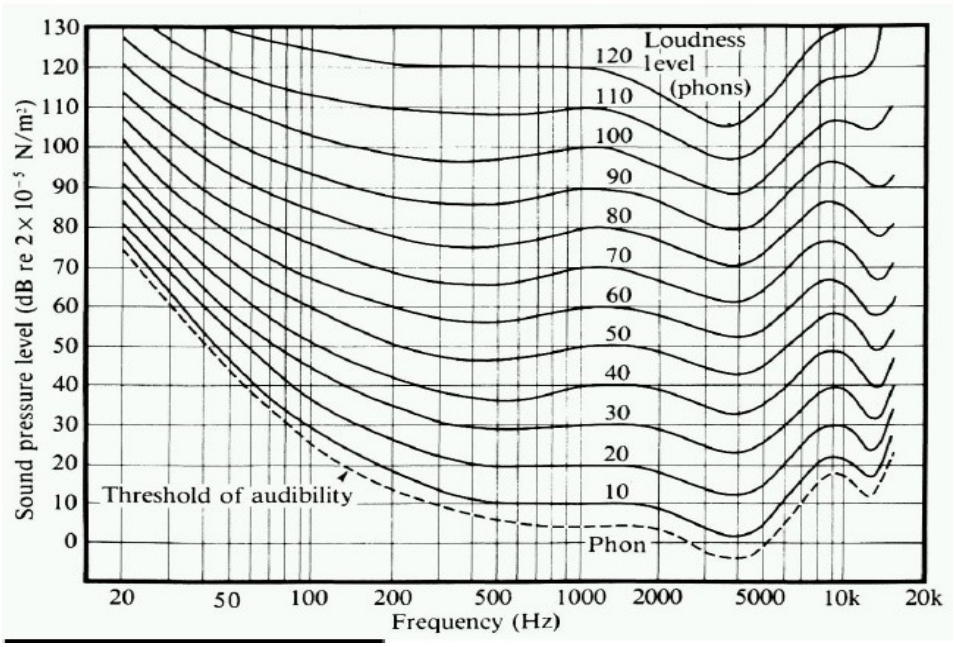


Figure 2.3: Equal-loudness contours for the human ear, determined experimentally by Fletcher and Munson, (Image from: Fletcher, H. and Munson, W.A.: *Loudness, its definition, measurement and calculation*. Journal of the Acoustic Society of America 5, 82-108 (1933))

In (**fig 2.3**), we can see that the loudness level is measured in phons. The phon was proposed as a unit of perceived loudness level L_N for pure tones. The purpose of the phon scale is to compensate for the effect of frequency on the perceived loudness of tones [12]. By definition, 1 phon is equal to 1 dBSPL at a frequency of 1 kHz[13]. In other words, a sine tone at 1kHz with intensity of 50dB has a loudness level of 50phons. Therefore, if we want, for instance, to produce a sine tone at 300 Hz with the same loudness as the 1kHz tone, all we have to do is follow the 50 phons curve until 300 Hz and use the corresponding value of SPL, then the two tones will sound equally loud to the listener.

Anatomy and function of the human auditory system

At this point, let us describe the basic aspects of the anatomy of the human ear and its function. Various sources contain information about the human auditory system anatomy, including [4], [9] and [14]. The peripheral auditory system is shown in (**fig. 2.4**). It is the medium by which sound waves are detected, encoded, and retransmitted through nerve cells to the brain, where sound as we know it is actually understood. Although very sophisticated, the process can be intuitively subdivided into three steps, each executed in a different part of the ear.

• The outer ear: amplifies and conveys incoming sound waves (air vibrations).
Here the sound waves enter the auditory canal, which can amplify sounds containing frequencies in the range between 3Hz and 12kHz. At the far end of the auditory canal is the eardrum (or tympanic membrane), which marks the beginning of the middle ear

• The middle ear: transduces air vibrations into mechanical vibrations.
Sound waves, coming from the auditory canal, are now hitting the tympanic membrane. Here, three delicate bones (in fact, they are the smallest bones of the human body), the malleus (hammer), incus (anvil) and stapes (stirrup), convert the low-level pressure eardrum sound vibrations into higher-level pressure sound vibrations to another, smaller membrane, called the oval or elliptical window. Finally, the smallest skeletal muscle of the human body, the stapedius muscle is there to stabilize the stapes, in order to prevent damages in the inner ear.

The middle ear still contains the sound information in wave form; it is converted to nerve impulses in the cochlea. Higher pressure is necessary because the inner ear beyond the oval window contains liquid rather than air.

• The inner ear: processes mechanical vibrations and transduces them mechanically, hydrodynamically and electrochemically (in this order). They are then transmitted through nerves to the parts of the brain responsible for their cognition.

The inner ear consists of the cochlea (**fig. 2.5**) and several non-auditory structures (the latter constitute the vestibular system, dedicated to balance). The cochlea has three fluid-filled sections, and supports a fluid wave driven by pressure across the basilar membrane separating two of the sections. Strikingly, one section, called the cochlear duct or scala media, contains an extracellular fluid similar in composition to endolymph, which is usually found inside of cells. The organ of Corti[3] is located at this duct, and transforms mechanical waves to electric signals in neurons. The other two sections are known as the scala tympani and the scala vestibuli. These are located within the bony labyrinth which is filled with fluid called perilymph. The chemical difference between the two fluids (endolymph & perilymph) is important for the function of the inner ear.

3.There are small hair cells along the basilar membrane which are connected with numerous nerve endings for the auditory nerves. These transmit information to the brain via a complex system of neural pathways. The hair cells come in four rows, and form the organ of Corti on the basilar membrane.

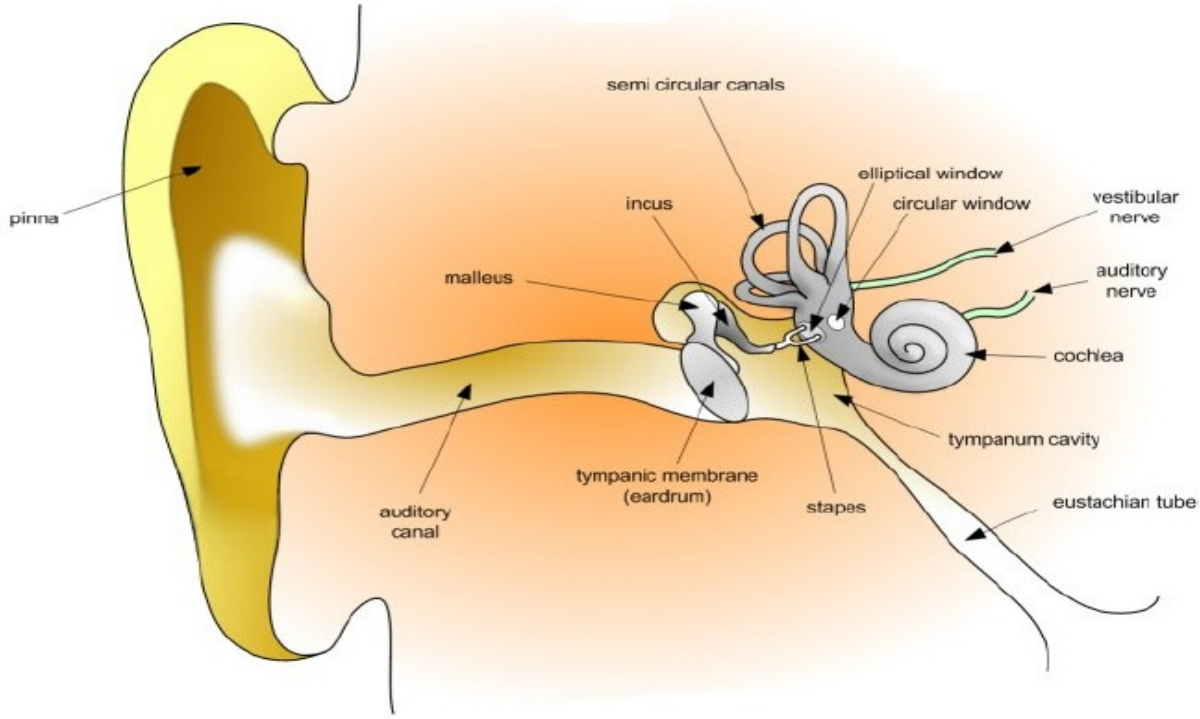


Figure 2.4: The human peripheral auditory system. (Image from: http://en.wikibooks.org/wiki/File:HumanEar.jpg)

When a sound wave reaches the ear, it is focused into the meatus, where it vibrates the ear drum. This causes the hammer, anvil and stapes to move as a system of levers, and so the stapes alternately pushes and pulls the membrana tympani secundaria in rapid succession. This causes fluid waves to flow back and forth round the length of the cochlea, in opposite directions in the scala vestibuli and the scala tympani, and causes the basilar membrane to move up and down.

When a pure sine wave is transmitted by the stapes to the fluid inside the cochlea, the speed of the wave of fluid in the cochlea at any particular point depends not only on the frequency of the vibration but also on the area of cross-section of the cochlea at that point, as well as the stiffness and density of the basilar membrane. For a given frequency, the speed of travel decreases towards the apical end, and falls to almost zero at the point where the narrowness causes a wave of that frequency to be too hard to maintain. Just to the wide side of that point, the basilar membrane will have to have a peak of amplitude of vibration in order to absorb the motion. Exactly where that peak occurs depends on the frequency. So by examining which hairs are sending the neural signals to the brain, we can ascertain the frequency of the incoming sine wave[4]. Pitch is the perceived parameter related to frequency, it can be thought as the quality of a sound, governed by the rate of vibrations produced by the sound [16].

Since real sounds have no single frequency, this region will show a place where excitation has a maximum, corresponding to the fundamental frequency. The distance of this maximum from the end of the basilar membrane is directly related to frequency, so that each frequency is mapped in a precise place along the membrane. The mechanical properties of the cochlea (wide and stiff at the base, narrower and much less stiff at the apex) denotes a roughly logarithmic decrease in bandwidth as we move linearly away from the cochlear opening (the oval window) as shown in (**fig. 2.5**). Thus, the auditory system acts as a spectrum analyzer, detecting the frequencies in the incoming sound at every moment in time. In the inner ear, the cochlea can be understood as a set of band-pass filters, each filter letting only frequencies in a very narrow range pass. This mechanism could be associated to a filterbank of constant-Q filters, because of their property to be linearly spaced on a logarithmic scale.

In fact, we will experiment in the last section of **chapter 5** with some fractional octave filterbanks, a subcategory of the constant-Q filter banks, with encouraging results.

The extent to which the ear can discriminate between frequencies very close to each other is not completely explained by the passive mechanics of the cochlea alone. More recent research shows that a sort of psychophysical feedback mechanism sharpens the tuning and increases the sensitivity. In other words, there is information carried both ways by the neural paths between the cochlea and the brain, and this provides active amplification of the incoming acoustic stimulus. The outer hair cells are not just recording information, they are actively stimulating the basilar membrane. One result of this feedback is that if the incoming signal is loud, the gain will be turned down to compensate. If there is very little stimulus, the gain is turned up until the stimulus is detected[4].

The sounds that the ear can sense have a wide frequency range, approximately from 20 Hz to 20 KHz. The perceived pitch, also expressed in Hz, has a limited range, approximately from 60 Hz to 5 KHz.

Additional processes occur at the brain level, for example, non-auditory neural information is used in order to combine signals coming from both ears and fuse them into one sensation. However, although complex, the mechanism does not yield enough information to the brain to allow it to understand, for example a single note, a harmony, a rhythm, or higher-level musical structures. Thus the nature of a sound is not only determined by the physical properties of sound and the human ear, but all this information will be combined at a higher level (i.e. in the brain) where the sound takes its musical form.

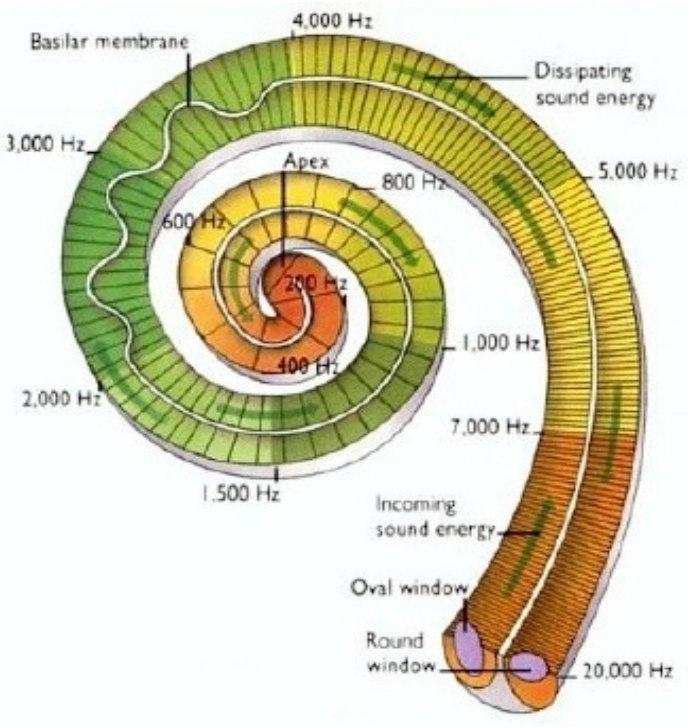


Figure 2.5: Part of the inner ear, the cochlea is shaped like a 32 mm long snail and is filled with two different fluids separated by the basilar membrane. (Image from: http://scienceblogs.com/startswithabang/2010/05/dont_you_hear_that.php)

4.An annoying side effect of this is that if mechanical damage to the ear causes deafness, then the neural feedback mechanism turns up the gain until random noise is amplified, so that singing in the ear, or tinnitus results.

Critical bands of hearing

Since each frequency stimulates a region of the basilar membrane, a limit to frequency resolution of the ear is imposed. This limit is reflected to another characteristic of perception, known as critical band. A simple example to understand how the ear works in the critical band is necessary. Think, or better listen, two sine waves very close in frequency, they have a total loudness which is less than the sum of the two loudness we would hear if they were separated in frequency. Now, if we slowly separate each other in frequency, we perceive the same loudness up to a point, then, over a certain frequency the total loudness increases approximately to the value of the sum of individual loudness. The frequency difference, needed to perceive loudness as sum of individual loudness is the critical band. The division of the frequencies into critical bands is the cause of other important factors of perception, such as auditory roughness.

Roughness is a sensation of dissonance, its presence is particularly strong in the lower and upper bound of the critical band, where the two tones are almost separated but not yet ready to be perceived as two sounds. In the middle of the critical band the two tones are heard as one with a frequency that lies between the two frequencies, where we can clearly perceive the sensation of beating. When the two tones are separated by 1 Hz we perceive a single beating per second. The width of critical bands (bandwidths) increase in frequency. In **chapter 5**, we will use roughness as a feature in our feature extraction stage with particularly good results.

The Bark scale

In order to represent the human ear behavior inside the critical bands, the Bark scale was proposed [17]. The Bark scale (of human hearing) ranges from 1 to 24 Barks, corresponding to the first 24 critical bands. The proposed Bark center frequencies, in Hz, are:
50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500, 13500

while their corresponding bandwidths, are:

100, 100, 100, 100, 110, 120, 140, 150, 160, 190, 210, 240, 280, 320, 380, 450, 550, 700, 900, 1100, 1300, 1800, 2500, 3500, 5000

These center-frequencies and bandwidths should be interpreted as being associated with a specific fixed filter bank in the ear. As we will see in **chapter 5**, during the extraction of some of our features we apply filters on the music signal based on the Bark Scale.

Timbre

Another perceptual attribute of hearing is timbre. Timbre is roughly defined as the attribute by which we can distinguish two sounds with the same loudness and pitch. Thus timbre is the character or quality of a musical sound, distinct but influenced from its pitch and loudness. Sometimes, it is also referred to as the color of sound. The characteristics determining timbre reside in the constantly changing spectrum of a musical sound, produced for example by an instrument. The steady-state spectrum is not enough to distinguish a sound produced by an instrument to another, but also the attack and decay portion of the spectrum are very important. Therefore, timbre has to be more than one dimension, because involves temporal envelope and evolution of the spectral distribution over time [16].

As timbre-related features we consider the auditory roughness we already mentioned above and the Mel-Frequency Cepstral Coefficients (MFCCs) based features. We will explore their merit in **chapter 5**.

Auditory masking effect

Sometimes the perception of one sound is affected by the presence of another sound. This effect is named auditory masking [18]. There are two kinds of auditory masking: (i) frequency masking or spectral masking and (ii) temporal masking. The first kind, is also called simultaneous masking as it is occurs between two concurrent sounds and it is often observed when the sounds share a frequency band. It has been shown that an intense sound of a lower pitch prevents us from perceiving a weaker sound of a higher pitch, but an intense sound of a higher pitch never prevents us from perceiving a weaker sound of a lower pitch. The explanation of this is that the excitation of the basilar membrane caused by a sound of higher pitch is closer to the basal end of the cochlea than that caused by a sound of lower pitch. So to reach the place of resonance, the lower pitched sound must pass the places of resonance for all higher frequency sounds. The movement of the basilar membrane caused by this interferes with the perception of the higher frequencies.The second kind of masking, also called non-simultaneous masking occurs when a sudden stimulus sound makes inaudible other sounds which are present immediately preceding or following the stimulus.

2.1.3 Sound as a Signal

Signals

In physics and engineering, a signal is a representation of a time-varying or spatial-varying physical quantity. Mathematically, a signal is sequence. A type of signals studied extensively is representations of time-varying quantities and are also referred to as time-series. They are so commonly encountered, that generally when we see the term 'signal' we usually imply it is a time-varying one.

A continuous-time real signal is any real-valued function which is defined for all time **t** in an interval, most commonly an infinite interval. If for a signal, the quantities are defined only on a discrete set of times, we call it a discrete-time signal. In other words, a discrete-time real signal can be seen as a function from (a subset of) the set of integers to the set of real numbers. If in addition to being a discrete-time real signal the signal also takes (amplitude) values that belong to a finite set, then it is called a digital signal. On the other hand, if it is a continuous-time signal takes (amplitude) values that belong to the set of real numbers it is called an analog signal

Since a sound is a continuous vibration of a medium (such as air), a sound signal associates a pressure value to every value of time and three space coordinates. It can therefore be considered viewed as a signal.

A microphone converts sound pressure at some place to just a function of time, so we need not bother with the space coordinates any more. So, to summarize we model sound as a continuous time-varying signal.

Analog signal to digital

In nature we encounter continuous signals very commonly, however as our computers and digital electronics in general only work on a basis of discrete values. We need to convert these signals into digital (that is discretized both in time and in amplitude). **Fig. 2.6** shows the necessary steps of this procedure.

To convert an analog signal to digital we first convert it to a discrete-time signal via sampling, that is by taking only particular samples of the signal a sound signal for example, we reduce it to a sequence of samples. Sampling is generally performed by measuring the value of the continuous signal every T seconds. he sampling frequency or sampling rate f_s is defined as the number of samples obtained in one

second, or $f_s = \frac{1}{T}$. The sampling rate is measured in Hz or in samples per second.

When it is necessary to capture audio covering the entire 20–20,000 Hz range of human hearing, such as when recording music or many types of acoustic events, audio waveforms are typically sampled at 44.1 kHz.

The next step is to convert the discrete-time signal into a digital signal. To do this we perform quantization on the discrete-time signal. Quantization can be roughly defined as the process of mapping a large set of input values to a smaller set.

Audio is typically recorded at 8-, 16-, and 20-bit depth, which yield a theoretical maximum signal to quantization noise ratio (SQNR) for a pure sine wave of, approximately, 49.93 dB, 98.09 dB and 122.17 dB [19].

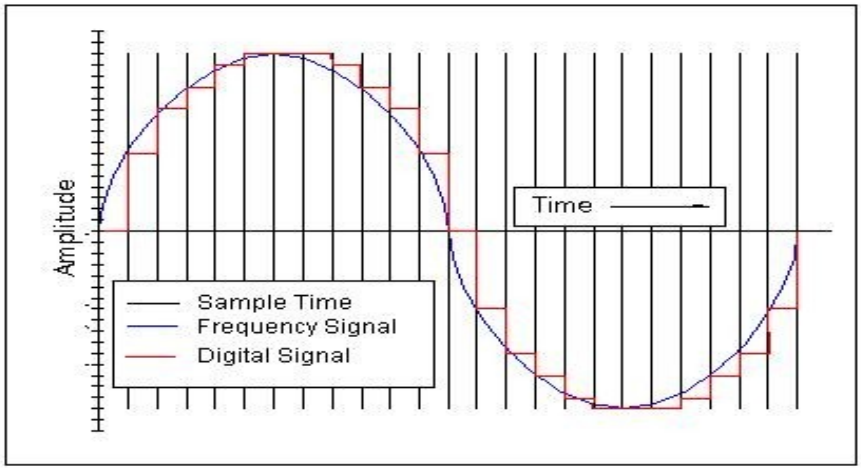


Figure 2.6: From analog signal to digital. First we convert the analog signal to a discrete-time signal via sampling, and then, to digital via quantization (Image from: http://www.soneti.net/csp/digital%20signal.JPG)

In theory, sampling is lossless. The initial signal can be reconstructed exactly as long as Nyquist–Shannon sampling theorem, which provides a sufficient (but not always necessary) condition under which perfect reconstruction is possible. The sampling theorem guarantees that band-limited signals (i.e., signals which have a maximum frequency) can be reconstructed perfectly from their sampled version, if the sampling rate is more than twice the maximum frequency. Quantization however is lossy. If a value of 5.14432 is converted into a 5 there is no way of tracing it back afterward.

Sometimes, each sample contains data for more than one channel. For example 2: a left and right channel, which may be considered to be a 2-vector signal, or equivalently 2 simple (one vector signals). In the general case of more than 1 channel, the sound is referred to as 'stereophonic' or 'stereo' sound. It is an attempt to create an illusion of directionality and audible perspective.

Fourier Transform

So when recording a song for example, the microphone after converting the (multidimensional) sound pressure to just a function of time, it generates a voltage signal as an analog representation of the sound signal. Now that we have established this, let us move on to what information we can derive from such a signal.

In order, for example, to look for specific notes of a song modeled as a digital signal, we must move from the time domain to the frequency domain, as a specific note corresponds to a specific pitch or

equivalently specific frequency. We therefore have to study closely the (frequency) spectrum of the signal. The frequency spectrum of a time-domain signal is a representation of that signal in the frequency domain. Any signal that can be represented as an amplitude that varies with time has a corresponding frequency spectrum.

We get the frequency spectrum of a signal by transforming it via the Fourier Transform. For a digital signal stored in a computer, the Discrete Fourier Transform (DFT) is applied to the signal through some kind of Fast Fourier Transform (FFT) algorithm.

Let us assume we have a discrete signal (e.g. a digital one), where we take N samples, denoted $f[0],\dots,f[N-1]$, where the samples are equally spaced in time. We define the Discrete Fourier Transform (DFT) as:

$$F[n]=\sum_{k=0}^{N-1} f[k]\mathrm{e}^{-j\frac{2\pi}{N}nk},\forall n\in[0,N-1] \quad \textbf{(Eq. 2.1.9)}$$

$F[n]$ is called the frequency spectrum or just spectrum, or DFT of the signal $f[k]$. We symbolize the transform by writing:

$$F[n]=\boldsymbol{F}\{f[k]\}$$

The Inverse Discrete Fourier Transform (IDFT) transforms a spectrum back to the time-domain signal from which it originated:

$$f[k]=\frac{1}{N}\sum_{n=0}^{N-1} F[n]\mathrm{e}^{+j\frac{2\pi}{N}nk} \quad \textbf{(Eq. 2.1.10)}$$

We symbolize the inverse transform by writing:

$$F[n]=\boldsymbol{F}\{f[k]\}$$

Often we will encounter Fourier Transform Pairs, that is paired time domain and frequency domain representations of a signal in a form similar to the one below:

$$\{F[n]\}\leftrightarrow\{f[k]\}$$

Filtering

In order to emphasize, de-emphasize, remove, or otherwise affect a component of a signal we use filters, devices or processes that do just that. Usually we apply filtering in the frequency domain and the target is to remove certain frequencies from the signal.

A filter in the time domain is modeled by its impulse response. In signal processing, the impulse response, or impulse response function (IRF), of a dynamic system is its output (hence it is a signal as well) when presented with a brief input signal, called an impulse, or the Dirac delta function, or δ function, defined as follows:

$$\delta(x)=\begin{cases}+\infty, x=0\\ 0, x\neq 1\end{cases}$$

Where the following identity must be also satisfied:

$$\int_{-\infty}^{\infty}\delta(x)\,dx=1$$

The application of a filter with impulse response $h(n)$ on a signal $x(n)$ in the time domain is equivalent to convolving the two signals. For discrete time, convolution takes the form of the following sum:

$$s(n)=x(n)*h(n)=\sum_{k=-\infty}^{\infty} x(k)h(n-k)=\sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

In the frequency domain, the filter is modeled by a transfer function $H(k)$. The application of a filter on a signal in the frequency domain is equivalent to multiplying the signal's spectrum with the filter's transfer function:

$$S(k)=H(k)\times X(k)$$

Convolution in the time domain is equivalent to multiplication in the frequency domain and vice-versa.

During the calculation of a number of features we apply various filters or filterbanks on the music signal, as we will see in chapter 5. A filter bank is simply an array of filters.

Features of a signal

An important attribute of a discrete signal $x[n]$ is its energy E_x :

$$E_x=\sum_{n=0}^{N-1}|x[n]|^2$$

Since we often think of signal as a function of varying amplitude through time, a good measure of the 'strength' of a signal would be the area under its curve. However, this area may have a negative part. That is why what we call the energy of a signal is actually the area under the squared signal.

Building on the above definition, we can also define the average power P_x of the discrete signal $x[n]$ as the energy per sample:

$$P_x=\frac{E_x}{N}=\frac{1}{N}\sum_{n=0}^{N-1}|x[n]|^2$$

This is useful when the energy of the signal goes to infinity. For example a periodic signal, does not decay. It has an infinite area under its curve, therefore infinite energy. A signal's power is a good measure of its 'strength' in such cases.

Parseval's theorem states that the area under the energy spectral density curve (in other words the area under the square of the magnitude of the spectrum $X[k]^2$) is equal to the area under the square of the magnitude of the signal $x[n]^2$, the total energy:

$$\sum_{n=0}^{N-1}|x[n]|^2=\frac{1}{N}\sum_{k=0}^{N-1}|X[k]|^2$$

Finally, another important feature of a signal is its spectrogram. A spectrogram is a time-varying spectral representation [20] that shows how the spectral density of a signal varies with time. It represents a signal in a joint time-frequency domain [21] and it also has the property of being positive. Usually the frequency domain is divided into bands and different gradations of color represent different values of the spectral density per time-(frequency band) bin. An example of a spectrogram is shown in **fig 2.7**.

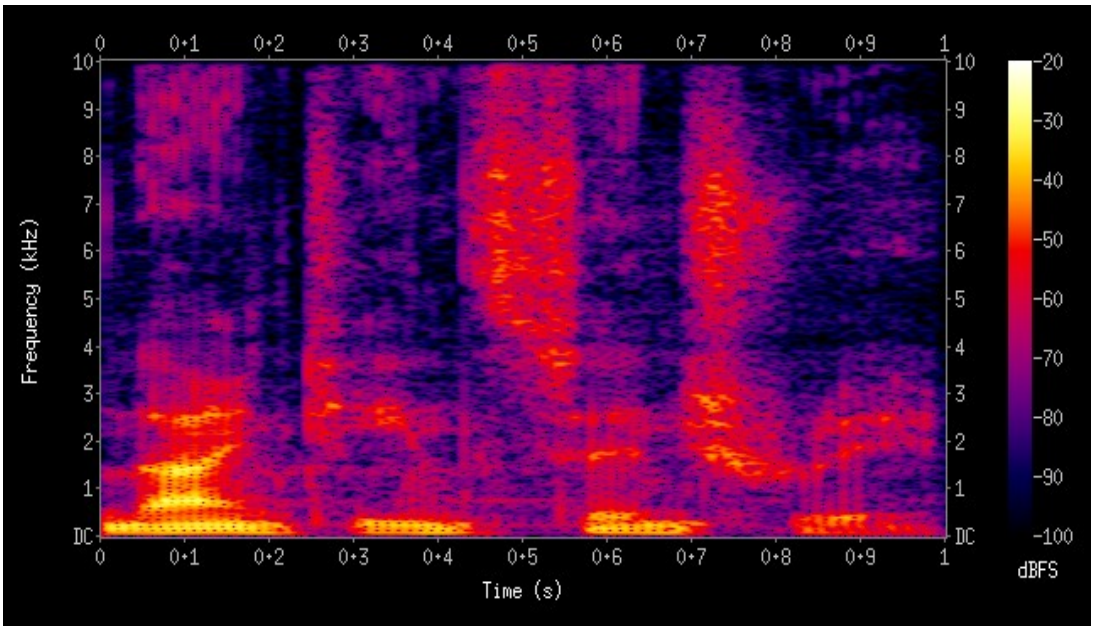


Figure 2.7: The spectrogram of a male voice singing. The horizontal axis represents time, the vertical frequency and the different colors correspond to different values of spectral density (here in dB) (Image from: http://en.wikipedia.org/wiki/File:Spectrogram-19thC.png)

Short-term features of a signal

Short-term features are estimated on a frame basis, we break down the signal into fragments called frames f_s through a process called 'windowing':

$$f\,s(n;m)=s(n)\,w(n-m)$$

where $s(n)$ is the initial signal and $w(m-n)$ is a window of length Nw ending at sample *m*.

The frames are usually overlapping, that is, a signal of length Ns is divided into more than Ns/Nw frames.

2.1.4 Music Theory Outline

Now let us take a look inside the inner workings of music and the relations underlying the various musical constructs such as notes, chords, keys, scales. The field concerned with the mathematics of music, (among other things) is Music theory. Music theory is the study of how music works. It examines the language and notation of music. It seeks to identify patterns and structures in composers' techniques, across or within genres, styles, or historical periods. In a grand sense, music theory distills and analyzes the fundamental parameters or elements of music-rhythm, harmony (harmonic function), melody, structure, form, texture, etc. [22] We will not focus that much on notation or on very high-level or exotic aspects of music theory. We will concentrate on modern western music and focus on the principles that govern the features studied in this work, so as to that in chapter 5 where we will discuss them, we can do so without any ambiguity.

Notes

A pure tone is a sound with a sinusoidal wave-shape (**fig. 2.9**). A sine wave is characterized by its frequency *ƒ*, the number of cycles per second -or its wavelength *λ*, the distance the waveform travels

through its medium within a period T - and the amplitude A, the size of each cycle. A pure tone has the unique property that its wave-shape and sound are changed only in amplitude and phase by linear acoustic systems.

$$v(t) = A\sin(2\pi ft) = A\sin(\omega t)$$

Where $\omega = 2\pi f$ is the angular frequency.

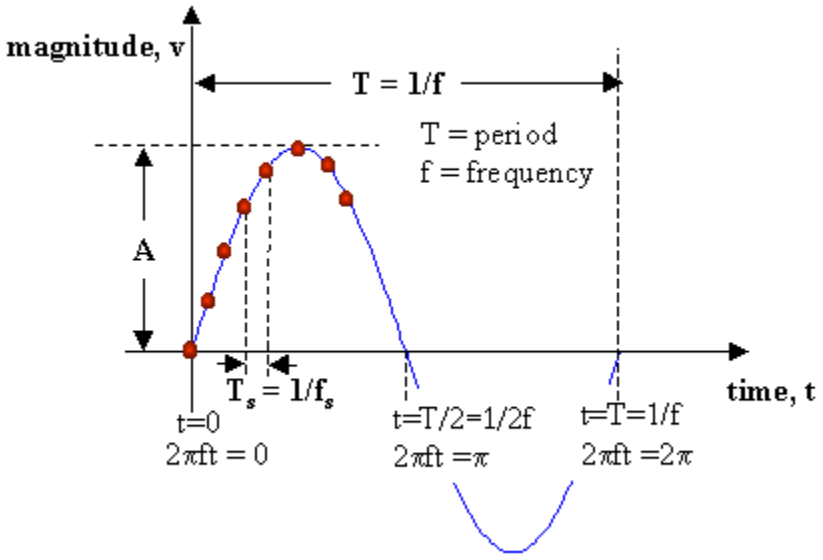


Figure 2.9: A sinusoidal (sine) wave, depicted in one period. The image also shows samples taken from it every Ts (sampling period).(Image from: http://users.rowan.edu/~shreek/networks1/music.html)

We already mentioned that sole pure tones are not generated by the vibrations in nature. The Fourier theorem states that any periodic waveform can be approximated as closely as desired as the sum of a series of sine waves with frequencies in a harmonic series and at specific phase relationships to each other. The lowest of these frequencies (called the fundamental frequency), which is also the inverse of the period of the waveform, determines the pitch of the tone, which is perceived by the human hearing. In music, notes are assigned to tones with different fundamental frequencies, in order to describe the pitch of played tones.

In other words, playing a note on a piano with a fundamental frequency of 220Hz does not produce a single sine wave of frequency 220Hz. Instead it also produces many other sine waves as well and the end result sounds quite pleasant. If we were to hear a single sine wave (e.g. produced electronically), it would sound discomforting, even downright annoying.

From now on, when we refer to a 'tone' we will mean a particular pitch, a particular note. There are 12 distinct tones in contemporary western music, 7 basic tones and 5 intermediate.

Let us start with the tone at 440Hz as reference[5]. All twelve tones are encountered between 440Hz and 880Hz at equal intervals called semitones or half-steps. Obviously, two half-steps constitute a step.

Sounds with a 2^n:1 ratio of frequencies on a just interval[6] are considered to have the same 'tonality' but they belong to a different octave. We can see certain 'logarithmic' attributes of music arise from this

5.In fact this frequency is usually treated as a frequency of reference. The A above middle C is usually set at 440 Hz (often written as "A = 440 Hz" or sometimes "A440")

6.A just interval or just intonation (sometimes abbreviated as JI) is any musical tuning in which the frequencies of notes are related by ratios of small whole numbers.

fact, which should not surprise us. After all, human hearing as we saw in section 2.1.3 works on a logarithmic basis as well.
So, to return to our previous example, at 880Hz we encounter the same tone (they share the same name: A) as the one at 440Hz, but an octave higher, since they have a 2:1 frequency ratio. In order to differentiate between the two, sometimes we write: $\mathbf{A_4}$ and $\mathbf{A_5}$. This notation is called 'scientific pitch notation' [23].

The basic tones have the following names (according to the two most common notations):

A	**La**
B	**Ti (Si)**
C	**Do**
D	**Re**
E	**Mi**
F	**Fa**
G	**So(l)**

Table 2.3: Note names in the two most common notations

The distances between them are the following (measured in semitones):

Transition	Distance in semitones
A → B	**2**
B → C	**1**
C → D	**2**
D → E	**2**
E → F	**1**
F → G	**2**
G → A	**2**

Table 2.5: Distances in semitones between consecutive notes

We use a sharp (♯) symbol to convert the note that follows to one semitone higher. For instance an **E**♯ is a semitone higher than **E**, that is, it is the note **F**.

Likewise, we use a flat (♭) symbol to convert the note that follows to one semitone lower. For instance an **E**♭ is a semitone lower than **E**, that is, it is the note **D**♯[7].

We can now 'fill the 2 semitone gaps' of the table above with notes using the symbolism of flats and sharps. So, finally, the 12 tones of an octave we mentioned at the beginning are the following, each a semitone apart from the preceding one:

7.Two notes, that like **E** ♭ **and D**♯ have a different name, but correspond to the same same sound (tonality) are called enharmonic notes.

Using Sharps Only	**A**	**A♯**	**B**	**C**	**C♯**	**D**	**D♯**	**E**	**F**	**F♯**	**G**	**G♯**
Using Flats Only	**A**	**B♭**	**B**	**C**	**D♭**	**D**	**E♭**	**E**	**F**	**G♭**	**G**	**A♭**

Table 2.6: The 12 tones of an octave

To better understand all this, let us take a look at a piano keyboard in figure 2.10.

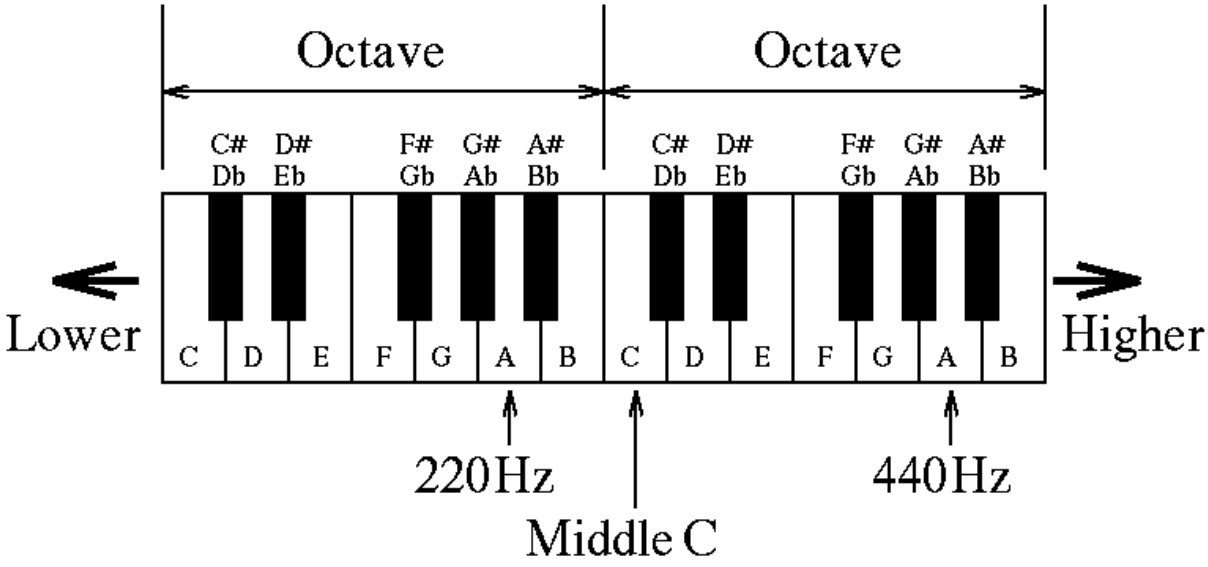


Figure 2.11: Two octaves of a piano keyboard. We can see 12 keys in each octave, each corresponding to a note, each a semitone apart from the previous one. The black keys are the sharps/flats. We can see their two notations above them. We can also notice the position of the middle C relative to the A at 440Hz. Finally, we can see the 2:1 frequency ratio between A4 and A5, two notes exactly an octave apart.
(Figure taken from: http://www.josef-k.net/mim/ThePianoKeyboard.gif)

Chroma

Chroma, is a term that is equivalent to tone. Two notes an octave apart belong to the same chroma class (the same pitch class), therefore, we have 12 chroma classes: The ones shown on table 2.6.

Scales

The musical scale is consisted of seven notes all in a row, in alphabetical order. If you count the first note, repeated an octave higher at the top of the scale, it is eight notes. This, in other words, means that a scale is, in fact, 8 (or 7 if we don't count the first note played an octave apart twice) successive pitches within a one-octave range. Together these notes are used to create melodies, as they generally 'sound good together', with 'good' not necessarily meaning 'happy' or 'calming', of course.

There are many different kinds of scales. The most common one is called the major scale. Major scales are 'happy' scales. They have pleasant and expected intervals at every turn. The mirror image of the major scale is the minor scale. Minor scales are sad scales; the intervals between the notes sound a little depressing. Both major and minor scales can start on any note—from **A**♭ to **G**♯. No matter which note you start with, each scale has its own specific combination of intervals between notes. This is what defines it [24].

As we already mentioned, a scale consists of seven main notes. We can use numbers to describe the 7 main notes in any scale. The first note is numbered one, the second note is numbered two, and so on. This method of numbering actually describes the 7 degrees of a musical scale. There also are some distinctive musical names used in place of the numbers, mainly in more formal situations. The following table presents these formal degree names (note that the eighth note is the first or tonic played an octave apart):

DEGREES OF THE SCALE	
DEGREE	**NAME**
First (Root)	**Tonic**
Second	**Supertonic**
Third	**Mediant**
Fourth	**Subdominant**
Fifth	**Dominant**
Sixth	**Submediant**
Seventh	**Leading Note**
Eighth (Octave)	**Tonic**

Table 2.6: The degrees of the scale and their 'musical' names

Keys

In music theory, the term key is used in many different and sometimes contradictory ways. A common use is to speak of music as being 'in' a specific key, such as 'in the key of **C** major or in the key of **F**♯'. Sometimes the terms 'major' or 'minor' are appended, as 'in the key of **A** minor' or 'in the key of **B**♭ major'.

Although the concept of musical key can be a complicated subject when examined closely, broadly speaking the phrase 'in key of **C**' means that **C** is music's harmonic center or tonic (the first degree of the scale, or the root of the scale). Note that the letter-name '**C**' does not indicate a single specific pitch but rather all pitches with the letter name **C** (sometimes called a pitch class). The successive pitches are an octave apart (or equivalently 8 degrees of the scale or 12 semi-tones on the chromatic scale).

The terms 'major' and 'minor' further imply the use of a major scale or a minor scale. Thus the phrase 'in the key of **E** major' implies a piece of tonal music harmonically centered on the note **E** and making use of a major scale whose first note, or tonic, is **E**. Although the term 'key' is commonly used this way, actual music can rarely be described so simply. This overview of the term also makes many assumptions and may not hold true for all forms of music.

Intervals

An interval is the frequency ratio between two notes. Important intervals are those measured by small-number ratios, such as 1:1 (unison or prime), 2:1 (octave), 3:2 (perfect fifth), 4:3 (perfect fourth), 5:4 (major third) etc.

It is also common to compare interval sizes using the scale of cents. This is a logarithmic scale in which

the octave is divided into 1200 equal parts. In equal temperament[8], each semitone is exactly 100 cents. The size in cents of the interval from frequency f_1 to frequency f_2 is

$$n = 1200\log_2(\frac{f_2}{f_1})$$

Chords

A chord is a combination of three or more notes played together. Basic chords consist of just three notes, arranged in thirds, called a triad. Within a specific chord, the first note is called the root (even if the chord isn't formed from the root of the corresponding scale). The other notes of the chord are named relative to the first note, typically being the third and the fifth above the chord's root [24].

In most cases, the type of chord is determined by the middle note: the third. When the interval between the first note and the second note is a major third (two whole steps, that is) you have a major chord. When the interval between the first note and the second note is a minor third (three half steps) you have a minor chord. As with major and minor scales, major and minor chords sound different to the listener, because the intervals in the chords are slightly different. Scales can be seen as ' combining pitches in time', while chords as 'combining pitches in frequency'. Like the scales bearing the same names, major chords create a feeling of happiness, while minor ones have a touch of sadness in them.

Here is a list of the most common chords names, their intervals and the feelings they are associated to:

- Major chords consist of a root, a major third, and a perfect fifth. They sound happy.
- Minor chords consist of a root, a minor third, and a perfect fifth. They sound sad.
- Diminished chords consist of a root, a minor third, and a diminished (lowered) fifth. They have a kind of eerie and ominous sound [24].
- Augmented chords consist of a root, a major third, and an augmented (raised) fifth.

Chords can include more than three notes. The other notes we add to an existing triad are called extensions. Chord extensions are typically added in thirds. So the first type of extended chord is called a seventh chord because the seventh is a third above the fifth. The second would be the ninth chord, which adds a third above the seventh… and so on. We will not go further than the sevenths, but we will describe the most important seventh chords, as we experiment a bit with them in **chapter 5**:

- (Dominant) seventh chords consist of a root, major third, perfect fifth, and minor seventh. They create a sensation of tension
- Major seventh chords consist of a root, major third, perfect fifth, and major seventh. They sound 'sweet' [24].
- Minor seventh chords consist of a root, minor third, perfect fifth, and minor seventh. They too, sound 'sweet' [24].

There are other types of chords as well:

- Altered chords are standard chords, like the ones we described, but with one or more alterations added to them. The end result varies according to the alteration.
- Power chords[9] consist of just a root and a fifth (we omit the third). They contain 'a lot of raw

8.An equal temperament is a musical temperament, or a system of tuning, in which every pair of adjacent notes has an identical frequency ratio. As pitch is perceived on a roughly logarithmic scale, this means that the perceived "distance" from every note to its nearest neighbor is the same for every note in the system

9.Power chords are a key element of many styles of rock music[25].

power', according to [24].

- Finally, suspended chords, up a half step to a perfect fourth. This suspension of the second note of the triad is so wrong to our ears, we want to hear the suspension resolved by moving the second note down from the fourth to the third as quickly as possible. In fact, composers usually resolve suspended chords, specially at the end of a musical phrase [24]. We experiment a little with them as well in **chapter 5**.

And thus ends our journey in the world of music. We saw but the major sights of this vast world but they will be enough to build upon in the next chapters. We now move on to a brief discussion about emotions. This will be apparent when we look at the agreement statistics between the annotators that participated in our study in **chapter 3**.

2.2 Emotion

In order to automatically classify music based on emotion, we must first define emotion... or at least, establish what we will mean from now on using the term and how we 'classify', or even 'measure' it. Here we will present the difficulties that arise when studying emotions, as well as some methods of categorizing them or measuring them. These methods will be used in the annotation process described in Chapter 3 and their success will be analyzed then. First come the bad news...

2.2.1 Challenges of Emotion Categorization

Searching for a definition of emotion, the most satisfying one is perhaps this one: "Emotion is the complex psychophysiological experience of an individual's state of mind as interacting with biochemical (internal) and environmental (external) influences". The word 'complex' does not bode well... And in fact, emotion is too complex to be really understood, as is anything related to the function of the human brain.

And since it is not yet understood, there does not yet exist a unique model of how to evaluate it, as well. Should it be modeled as a discrete or a continuous quantity?

Some categorizations include:

- 'Cognitive' versus 'non-cognitive' emotions
- Instinctual emotions (from the amygdala), versus cognitive emotions (from the pre-frontal cortex)
- Basic versus complex: where base emotions lead to more complex ones
- Categorization based on duration: Some emotions occur over a period of seconds (e.g., surprise) where others can last for years (e.g., love)
- Categorization based on the three-dimensional space of activation (arousal), potency(power) or dominance, and valence (pleasure), or two-dimensional planes of this space.

Looking back at the definition we also encounter the word 'individual'. From our experience, we already know that emotion is defined in individual terms. The emotions two persons experience when presented with the same stimulus might be entirely different. Something to be expected, as microbiological factors, unique to each individual, come into play.

Finally, it is important to distinguish between three different 'types' of emotion: intended, expected and

experienced emotion [26].

- Intended emotion describes the emotional response that the song attempts to evoke in its listeners
- Experienced emotion describes the emotion a user actually feels when listening to the song
- Expected emotion is the expected value of experienced emotion in a population

Our main intention was to classify the song samples based on their intended emotion and the annotators were given explicit instructions to try to "think what emotion the composer of the song wanted to convey with it". However, it is impossible that the annotators have been totally objective, and their experienced emotion has almost certainly affected their annotation.

Keeping these difficulties in mind, a model had to be selected in order to move on to the next stages. In this work, we experimented with two methods of categorizing emotion: (i) 'Mood Clusters' and (ii) a two-dimensional representation of the emotion in the Valence-Activation plane.

2.2.2 Mood Clusters

In a first attempt, we used the method of 'mood clusters' used extensively from the Music Information Retrieval Evaluation eXchange (MIREX)[10] for Audio Music Mood[11] Classification, proposed by [27]. The main idea is that similar emotions are clustered together in five categories. This way we confine the space of the values of the emotions to just these five sets of emotions. The following categorization of mood labels is suggested:

- Cluster 1: passionate, rousing, confident, boisterous, rowdy
- Cluster 2: rollicking, cheerful, fun, sweet, amiable/good natured
- Cluster 3: literate, poignant, wistful, bittersweet, autumnal, brooding
- Cluster 4: humorous, silly, campy, quirky, whimsical, witty, wry
- Cluster 5: aggressive, fiery,tense/anxious, intense, volatile,visceral

2.2.3 Emotional Dimensions

Our second approach was to treat the emotion space as consisting of distinct components. In other words, we consider it a multi-dimensional space and study its component dimensions individually.

A three-dimensional space has been proposed [28] to describe emotions, its dimensions are:

- Valence (or 'pleasure-displeasure' or 'happiness-sadness')
- Activation (or 'arousal')
- Potency (or 'dominance-submissiveness')

The three dimensional system has been extensively use. However, it has been shown that a two-dimensional plane consisting only of Valence and Activation is adequate to represent the range of emotions experienced by listeners [29]. We study these two dimensions in this thesis.

10. More about MIREX on: http://www.music-ir.org/mirex/wiki/MIREX_HOME
11. Mood is loosely used to refer to the 'long term average o emotion'.

2.3 Classification

Our problem, automatically classifying music samples based on their intended emotion, is a classification task. We already analyzed the peculiarities of music and emotion, so now let us present the theory behind the classification problems, and discuss briefly the particular classifiers we will use in **Chapter 6**. Some details about the training of the classifiers and their evaluation will be left to be discussed in that chapter

2.3.1 Definition of a Classification Problem

In machine learning[12], classification is the problem of identifying the sub-population to which new observations belong, where the identity of the sub-population is unknown, on the basis of a training set of data containing observations whose sub-population is known. Thus the requirement is that new individual items are placed into groups based on quantitative information on one or more measurements, traits or characteristics, etc. and based on the training set in which previously decided groupings are already established.

In a more mathematical context, let us assume that we have two sets of data (datasets): the training set *T* and the testing set *D*.

The training set *T* consists of information x and y for each data-point, where x denotes what is generally a vector of observed 'characteristics' or 'features' for the data-item (we shall call it the data-item's 'feature vector') and y denotes a group-label (class). The label **y** can take only a finite number of values.

The classification problem can be stated as follows: given training data:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

produce a rule (or 'classifier') h, such that $h(x)$ can be evaluated for any possible value of x (not just those included in the training data) and such that the group attributed to any new observation, specifically

$$\hat{y} = h(x)$$

is as close as possible to the true group label y.

For the training set *T*, the true labels y_i are known but will not necessarily match their in-sample approximations

$$\hat{y}_i = h(x_i)$$

For new observations, the true labels $\mathbf{y}_j$ are unknown, but it is a prime target for the classification procedure that the approximation

$$\hat{y}_j = h(x_j) \approx y_j$$

as well as possible, where the quality of this approximation needs to be judged on the basis of the statistical or probabilistic properties of the overall population from which future observations will be drawn. The testing set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, whose true labels y_i are also known, is used during the evaluation of how well has the classification been performed. The measures we used to quantify this will be discussed in **Chapter 6**.

12. A classification problem is, in particular, a supervised learning task, as during the training step, we know beforehand the target of our classification, hence the desired output. In other words, our training data are supervised.

Multiple classification techniques have been developed, but classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems. In other words, the best classification technique is problem-specific and usually found through experimentation.

We used three classification methods in our work: (i) the Naïve Bayes classifier, (ii) the Multilayer Perceptron classifier and (iii) the k-nearest neighbor classifier. Let us examine each one of them more closely.

2.3.2 Classification Methods Used

Naïve Bayes Classifier

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (or 'naïve', hence its name) independence assumptions. A more descriptive term for the underlying probability model would be 'independent feature model'. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

Suppose we have a finite set C of possible classes c an item can belong to, and a feature vector consisting of n features: $F1$ through Fn. The model that describes that each class is dependent (or better: 'conditional') on the features consists of the conditional probabilities of the form:

$$p(C|F_1,\dots,F_n), \forall\, c\in C$$

Using Bayes theorem, we can equivalently write:

$$p(C|F_1,\dots,F_n)=\frac{p(C)\,p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}, \forall\, c\in C \quad \textbf{(Eq. 2.3.1)}$$

The individual components are:

- $p(C|F_1,\dots,F_n), \forall\, c\in C$: 'Posterior' or 'a-posteriori' probability. The probability that given the particular values of the feature, the item belongs in a certain class.
- $p(C), \forall\, c\in C$: 'Prior' or 'a-priori' probability. The probability that the item belongs to a certain class if no other information is available about it.
- $p(F_1,\dots,F_n|C), \forall\, c\in C$: 'Likelihood'. The probability that an item's features have the particular values, given that it belongs to a certain class.
- $p(F_1,\dots,F_n), \forall\, c\in C$: 'Evidence'. The probability that the item's features have the particular values if no other information is available.

In other words, **Eq. 2.3.1** means:

$$Posterior=\frac{Prior\times Likelihood}{Evidence}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant

We notice that: $p(C)\,p(F_1,\dots,F_n|C)=p(F_1,\dots,F_n,C), \forall\, c\in C$.

A joint probability can be rewritten using repeated applications of the definition of conditional

probability as:

$$p(F_1,\dots,F_n,C)=p(C)\,p(F_1,\dots,F_n|C),\forall c\in C$$
$$p(F_1,\dots,F_n,C)=p(C)\,p(F_1|C)\,p(F_2,\dots,F_n|C,F_1),\forall c\in C$$
$$p(F_1,\dots,F_n,C)=p(C)\,p(F_1|C)\,p(F_2|C,F_1)\,p(F_3,\dots,F_n|C,F_1,F_2),\forall c\in C$$
$$\vdots$$
$$p(F_1,\dots,F_n,C)=p(C)\,p(F_1|C)\,p(F_2|C,F_1),\dots,p(F_n|C,F_1,F_2,\dots,F_{n-1}),\forall c\in C \quad \textbf{(Eq 2.3.2)}$$

Now let us assume that each feature F_i is conditionally independent of every other feature F_j for $i\neq j$. This means that :

$$p(F_i|C,F_j)=p(F_i|C),\forall c\in C \qquad \textbf{(Eq 2.3.3)}$$

By **Eq 2.3.2** and **Eq 2.3.3** we get:

$$p(F_1,\dots,F_n,C)=p(C)\,p(F_1|C)\,p(F_2|C)\cdots,\forall c\in C$$

$$p(F_1,\dots,F_n,C)=p(C)\prod_{i=1}^{n}p(F_i|C),\forall c\in C \qquad \textbf{(Eq 2.3.4)}$$

So by **Eq 2.3.1** and **Eq 2.3.4** we get:

$$p(C|F_1,\dots,F_n)=\frac{1}{Z}\,p(C)\prod_{i=1}^{n}p(F_i|C),\forall c\in C \qquad \textbf{(Eq 2.3.5)}$$

Where Z (the evidence) is a scaling factor, a constant if the values of the feature variables are known.

Using this independent feature model and a common decision rule- 'pick the hypothesis that is most probable[13]'- the Bayes classifier is the following function:

$$classify(f_{1,\dots},f_n)=\underset{c}{argmax}\;p(C=c)\prod_{i=1}^{n}p(F_i=f_i|C=c)$$

An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because it assumes the variables are independent, only the variances of the variables for each class need to be determined and not the entire covariance matrix. Its disadvantages lie, of course in this assumption as well. If the features are not really independent, the 'independent feature model' fails to model correctly the problem.

In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. Moreover there do exist some theoretical explanations the surprisingly good performance of naive Bayes classifiers [30].

Multilayer Perceptron Classifier

The multilayer perceptron (usually abbreviated MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a

13.This is known as the maximum a posteriori (MAP) decision rule.

supervised learning technique called backpropagation for training the network.[31][32] MLP is a modification of the standard linear perceptron, which can distinguish data that is not linearly separable[33]. Let us clarify some things:

A neural network is a network of interconnected biological neurons. An artificial neural network is a computational model that mimics the way the biological neural networks are presumed to work. It is comprised of artificial neurons (**fig. 2.3.1**).

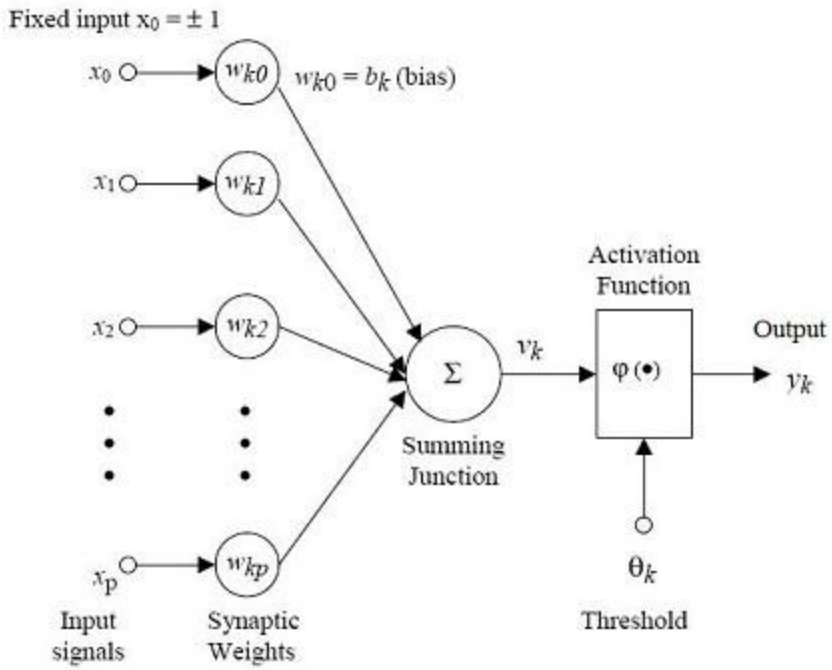


Figure 2.3.1: Model of an artificial neuron (Image from: http://www.learnartificialneuralnetworks.com/)

Based on the symbolism used in **fig. 2.3.1**, let us describe the basic function of an individual neuron.

The synapses of the biological neuron (the pathways which interconnect the neural network and give the strength of the connection.) are modeled as weights. For an artificial neuron, the weights (here: w_{k1} through w_{kp} are numbers, and represent the synapse. A negative weight reflects an inhibitory connection, while positive values designate excitatory connections.

The variables $x_{1,}x_{2,\ldots},x_p$ represent the input of the neuron, the product $x_0 \times w_{k0}$ represents a fixed bias.

All inputs are multiplied by their corresponding weight and the products are summed together (along with the bias). This activity is referred as a linear combination:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j$$

The output of the neuron, y_k, would therefore be the outcome of some activation function on the value of v_k. An acceptable range of output is usually between 0 and 1, or it could be -1 and 1 . It could be a thresholding function giving one of two discrete values, or a logistic function, giving values from a continuous interval.

As we said, neural networks are networks comprised of individual neurons. The output of some neurons becomes the input of others and so on. A 'feedforward' neural network is an artificial neural network where connections between the units do not form a directed cycle. Information always moves one direction, it never goes backwards.
The standard linear perceptron is the simplest kind of feedforward neural network: it is in effect a linear classifier. That is, it can only distinguish data that is linearly separable (the decision curve that separates the classes must be linear). We will not analyze it in more depth here. However, suffice it to say that the multilayer perceptron that utilizes the backpropagation technique is able to differentiate between any kind of data.

The multilayer perceptron (**fig. 2.3.2**) consists of three or more layers (an input and an output layer with one or more 'hidden layers') of nonlinearly-activating nodes. Each node in one layer connects with a certain weight w_{ij} to every node in the following layer.

Learning occurs in the perceptron by changing the connection weights after each new item is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning -as we already know the target output- and is carried out through the backpropagation technique.

The error in output node j in the n-th data point is $e_j(n)=d_j(n)-y_j(n)$, where d is the target value and y is the value produced by the perceptron. We then make corrections to the weights of the nodes based on those corrections which minimize the error in the entire output, given by:

$$E(n)=\frac{1}{2}\sum_j e_j^2(n)$$

Using gradient descent, we find our change in each weight to be:

$$\Delta w_{ji}(n)=-\eta\frac{\partial E(n)}{\partial v_j(n)}y_i(n)$$

where y_i is the output of the previous neuron and η is the 'learning rate', which is carefully selected to ensure that the weights converge to a response fast enough, without producing oscillations. In programming applications, this parameter typically ranges from 0.2 to 0.8.The derivative to be calculated depends on the activity v_j . For an output node this derivative can be simplified to:

$$-\frac{\partial E(n)}{\partial v_j(n)}=e_j(n)\Phi'(v_j(n))$$

where Φ' is the derivative of the activation function.

The relevant derivative for a hidden node is:

$$-\frac{\partial E(n)}{\partial v_j(n)}=\Phi'(v_j(n))\sum_k -\frac{\partial E(n)}{\partial v_j(n)}w_{kj}(n)$$

This depends on the change in weights of the kth nodes, which represent the output layer. So to change the hidden layer weights, we must first change the output layer weights according to the derivative of the activation function, and so this algorithm represents a 'backpropagation of the

activation function'. [34]

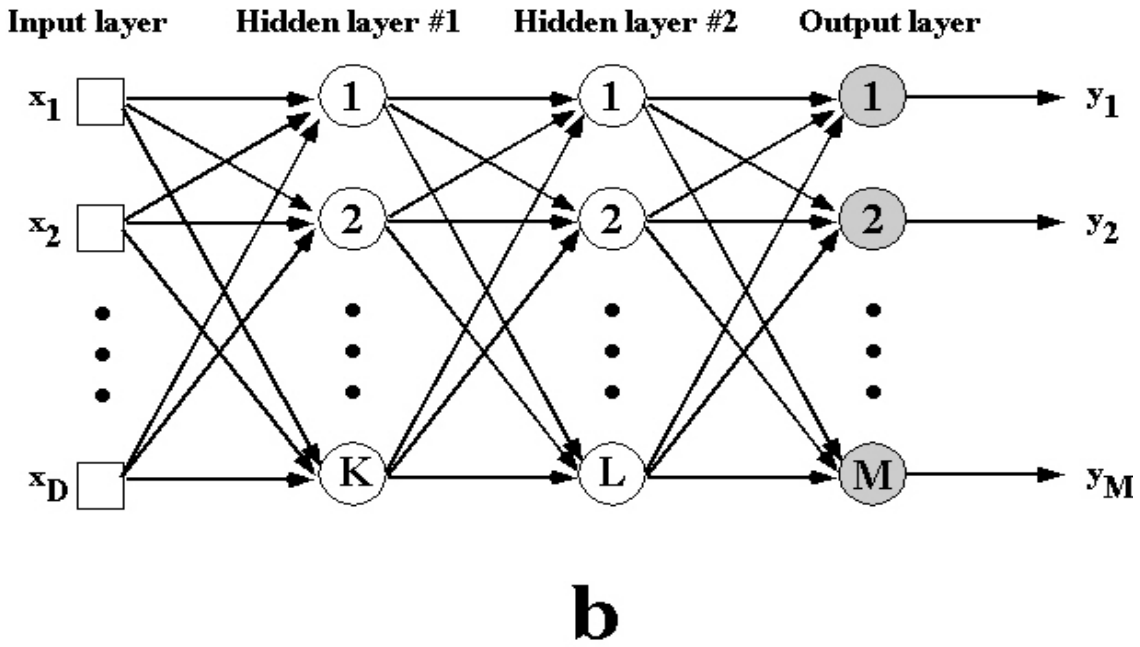


Figure 2.3.2: Generic depiction of the 2 Hidden Layer Perceptron (Image from: http://www.cimms.ou.edu/~schultz/snowdensity/paper.shtml)

The learning rate, the number of hidden layers, the activation function used, all these parameters will affect the performance of a multilayer perceptron classifier. We cannot know beforehand which values are the optimal for our particular problem and we must find them through experimentation.

k-Nearest Neighbor Classifier

The *k*-nearest neighbor algorithm[14] (usually abbreviated *k*-NN) is a method for classifying items based on closest training examples in the feature space. It is amongst the simplest of all machine learning algorithms: an item is classified by a majority vote of its *k* nearest neighbors, with the item being assigned to the class most common among them.

k is a positive integer, typically small, and usually odd to avoid ties. The best choice of *k* depends upon the data. Yet another case of 'the parameters of a successful classification technique, are problem-specific'. Generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. (they generalize better, or they overgeneralize). A good *k* is selected through the use of experimentation and heuristics.

At its simplest version, the *k*-NN algorithm is simple to implement, but very computationally demanding, especially for large training sets, as it entails many comparisons among the new item and the training samples. On the upside, it has some strong consistency results. As the amount of data approaches infinity, the algorithm is guaranteed to yield an error rate no worse than twice the 'Bayes error rate' (the minimum achievable error rate given the distribution of the data)[35].

This was the theoretical framework in which we will work. Additional theoretical information might be offered during the description of our work when deemed necessary.

14.k-NN is a type of 'instance-based learning', or 'lazy learning' where the function is only approximated locally and all computation is deferred until classification.

3. DATASETS, TOOLS USED & PRE-PROCESSING OF THE DATA

In this chapter we will discuss the first steps of our work. We will present the original data we used, and our efforts to create from it a data-set of songs in order to use during the annotation (**Chapter 4**) and the feature extraction (**Chapter 5**) steps. We will briefly present the tools and programming environments we used throughout our work. Finally, we will also make some references to the other modalities studied (lyrics, chords, EEG measurements) and the necessary pre-processing the data from them have undergone too, in order to become useable.

3.1 Datasets

In order to train a classifier and evaluate its performance we need examples. So our first step should be to obtain a data-set to use for training our models. This data-set should be annotated, or labeled, that is, the target output of the classifier should be known. Unfortunately, music classification being a relatively new domain of research, already-annotated data-sets are hard to come by, so we had to create our own based on already existing music collections, of course. We used two such collections:

(i) A collection of 240 classical music compositions of various artists.
(ii) A collection of 180 songs by The Beatles.

Both collections have already been extensively utilized in the past in music classification tasks. The second collection, had the added benefit of being accompanied by files (one for each song) recording the chords of the songs and the time intervals during which they were played. Moreover, when we decided to study a little the effect of the lyrics on the emotion, The Beatles songs' lyrics were easy to find.

3.2 Tools & Programming Environments

Now follows a brief presentation of the tools and scripting environments we utilized.

Matlab[36]
We worked mainly in matlab both for trivial tasks like batch processing files, visualizing data, parsing text files, converting from one format to the another etc., and for more complicated applications like creating a program with a graphical user interface to be used during the annotation process.

MIRtoolbox 1.3.1 [37]
During the feature extraction stage, we made extensive use of the MIRtoolbox, a Music Information Retrieval toolbox for matlab. MIRtoolbox is open source and it offers a wide array of musical feature extraction functions.

Weka 3.6.3 [38]
The tool we used throughout the entire classification process was Weka, an open source collection of machine learning algorithms for data mining tasks, designed by the Machine Learning Group at University of Waikato, New Zealand. It stands for 'Waikato Environment for Knowledge Analysis'. It is written in Java and offers a wide array of parametrized classifiers with many options regarding feature selection, visualization, classifier evaluation, etc.

Praat 5.2.28 [39] and openSMILE 1.0.1 [40]
These two programs were used to a far lesser extend in our work, mainly for experimenting reasons. They are both used for feature extraction. However, we opted to use matlab and MIRtoolbox, almost exclusively in our work.

Praat is an open source program used in analysis, synthesis, and manipulation of speech. It is written in C/C++. We experimented a bit with its functions, however we did not use it to extract any of the features presented in this thesis.

open*SMILE* combines features from Music Information Retrieval and Speech Processing. SMILE is an acronym for Speech & Music Interpretation by Large-space Extraction. It is written in C++ and it is also open source. The extraction of the Mel-frequency Cepstral Coefficients (MFCCs) was performed using an openSMILE script.

Audacity 1.3 Beta and Free MP3 Converter
These two programs were only used in some special cases, namely the conversion of sound files from one format to another and the creation of song samples from The Beatles' song collection.

We used Audacity[15] to listen to The Beatles' songs and isolate appropriate excerpts of them (see **section 3.3.1** for details about the procedure).

Free WAV to MP3 Converter[16] was use to convert .wav files from one format to another (from the high quality used for annotation, to the low quality used for feature extraction. See **section 3.3.1** for details about them.)

3.3 Pre-processing of the Data

Our first task was to create a data-set of song excerpts, or 'song samples', as we will most commonly refer to them throughout this thesis taken from the two musical collections we mentioned earlier. Some of the music compositions, especially in the classical music collection, were very large (some even more than half an hour-long!), so processing the entire compositions and extracting features from them would be impossible. Moreover, we also needed samples of a small duration, because emotion would undergo serious changes during a larger song, and so would the features studied, thus rendering both annotation and feature extraction meaningless.

3.3.1 Sound Signal Data

The .wav format
The original music collections comprised of music pieces stored in .wav format or 'Waveform Audio File Format', or 'WAVE'). The .wav format is actually the digital representation of a sound signal, as we described it in **section 2.1.3**.

The main attributes of a .wav file are the following:

- Duration: The duration of the sound sample (measured in seconds).
- Sampling Frequency or Sample Rate: The frequency at which the original (analog) sound signal

15 Available for free at: http://audacity.sourceforge.net/
16.Available for free at: http://www.free-audio-converter.net/free_wav_mp3_converter.html

was sampled in order to be transformed to a discrete signal (measured in Hz).

- Bits Per Sample: the number of bits used for determining the levels of the quantization of the original (discrete) sound signal in order to be converted to a digital signal. With b bits we can have up to 2^b quantization levels.
- Number of Channels: The actual number of vectors that make up the data contained in the file. Each vector corresponds to a different channel, in essence a different digital signal, and together (if they are more than 1) they create a perception of depth to the listener. It is usually either 1 channel (mono) or 2 channels (stereo).

Based on the above attributes, sometimes we especially refer to the number of bytes per second (or byte rate):

$$Byte\ Rate = \frac{Sample\ Rate \times Number\ of\ Channels \times Bits\ Per\ Sample}{8}$$

Or, bearing in mind that $1\,byte = 8\,bits$ we use the bits per second (or bit rate):

$$Bit\ Rate = Sample\ Rate \times Number\ of\ Channels \times Bits\ Per\ Sample$$

The .wav format consists of a header and a data part. We shall ignore the header part. The data part's size in bytes is:

$$Bytes = Byte\ Rate \times Duration = \frac{Sample\ Rate \times Number\ of\ Channels \times Bits\ Per\ Sample \times Duration}{8}$$

And in bits:

$$Bits = Bit\ Rate \times Duration = Sample\ Rate \times Number\ of\ Channels \times Bits\ Per\ Sample \times Duration$$

We made use of the above formulae in our code during various stages in cases we wanted to calculate the duration of a song, or to check the consistency of our data and/or song samples extracted from them by comparing their supposed duration to their estimated one and marking any discrepancies.

Selection of the Samples

We followed different strategies in the way we selected and extracted the song samples in each of the two collections described in the beginning of this chapter. Let us examine each case:

- Collection 1: 240 classical music compositions of various artists.

  We selected one sample of 20 seconds duration from each song. Each sample was randomly selected from the middle part of the song (between the ¼ and the ¾ of the song's total duration), so songs with $duration < 40\,seconds$ had to be discarded before this procedure.

  We applied a linear fade-in effect of 3 seconds in duration and a linear fade-out effect of 1 second in duration, so that the listeners of the song would not be startled by abrupt changes in the sound's volume.

  We then manually (a script played each song and asked us whether to discard it or not) discarded the samples containing human voice, notable fluctuations in their presumed intended emotion (at least to the point that this was possible) and long pauses. We ended up with 181 samples.

Finally, we renamed the songs to file1.wav, file2.wav, ..., file181.wav in order to hide their title and composer, as it might create unwanted expectations to the listener that might affect their judgment during annotation.

All of the above was done with appropriate matlab scripts we created.

- Collection 2: 180 songs by The Beatles.

  At this point, having learned some lessons from the previous procedure, and wanting to be able to explore other modalities as well (chords, lyrics), we turned to this song collection and also changed a bit our method of extracting the samples to a more 'manual' one.

  We used Audacity to listen to The Beatles' songs and isolate appropriate excerpts of them. We now chose 1 to 5 excerpts of each song of varying durations in the range of 10-20 seconds.

  Each sample was selected so that it met the following criteria. (i)It was fairly distinct from the other ones taken from the same song (usually we selected one from the beginning of the song, one from its chorus and one from the song's finale). (ii) Its emotional content was fairly the same throughout its duration. (iii) it did neither begin, nor end very abruptly both musically and in terms of lyrics.

  The process was copious, but the result was rewarding, as we ended up with 412 samples (more than the ones derived automatically from the first collection), and with better qualities (the ones implied by the three criteria above).

Samples' Format

In both cases, we used two different .wav configurations for our samples. The reason behind this choice, is that different steps of our task, demand different qualities from the samples. The samples to be used during the annotation stage have to be of as high sound quality (without, of course, exceeding the quality the ear can actually discern) as possible, in order for the listeners' emotional experience to be as distinctive as possible.

On the other hand, high quality entails more information (more samples per second, samples quantized to more levels so more bits are needed to represent amplitude, 2 channels are used therefore double the information). And more information leads to two problems: (i) more memory is needed to store it (memory complexity) and more operations, thus more time is needed to perform calculations using it (time complexity). And even if we do not care much about these issues during the annotation step, during the feature extraction stage it would be a serious problem. So for the feature extraction stage we sacrificed quality to guarantee ensure a good performance.

The .wav configurations actually used are the following:

- Annotation Format

  Sampling Frequency = 44.1 kHz
  Bits Per Sample = 16 bits/sample (signed 16 bit PCM)
  Number of Channels = 2 (stereo)
- Feature Extraction Format

  Sampling Frequency = 22.05 kHz

Bits Per Sample = 8 bits/sample (unsigned 8 bit PCM)
Number of Channels = 1 (mono)

3.3.2 Chord Data

The Beatles' songs were accompanied by their corresponding chords (one .txt file containing the chords of the .wav file of the same name). Before we could make any use of this data, they, too had to undergo a pre-processing stage.

First-off, we had to isolate the parts of the chords' files that corresponded to the song's samples we extracted. The format of the chords's files was the same as the following example:

(Start) (End) (Chord)
0.0000 3.9073 N
3.9073 10.6478 A:7
10.6478 14.0762 D:7
14.0762 17.5278 A:7
17.5278 20.9794 E:7
20.9794 24.4194 A:7
24.4194 31.2761 A:7
31.2761 34.7277 D:7
34.7277 38.1561 A:7
38.1561 41.6426 E:7
41.6426 45.4838 A:7
⋮

We implemented a matlab script that would ask for the beginning and ending time of a sample, and then it would find the chords file of the song from which it was taken and create a new file, bearing the name of the song sample and containing only the chords corresponding to it.

For example if we would choose from the song with the chords above the part between the 10thand the 30th second, we would get:

10.0000 10.6478 A:7
10.6478 14.0762 D:7
14.0762 17.5278 A:7
17.5278 20.9794 E:7
20.9794 24.4194 A:7
24.4194 30.0000 A:7

Next, we normalized the chord files, so that the times would correspond to the times of the sample and not the song from which they were taken. In other words, we subtracted from each time the starting time. Continuing the example above, we would get:

0 0.6478 A:7
0.6478 4.0762 D:7
4.0762 7.5278 A:7
7.5278 10.9794 E:7
10.9794 14.4194 A:7
14.4194 20.0000 A:7

Now the chords' files as well were ready to be used for feature extraction (**Chapter 5**).

3.3.3 Lyrics Data

The Beatles' songs contained lyrics and this time we thought it would be a good idea to experiment a bit with them as well. We did not have any other choice, but to create manually from scratch a data-set of the lyrics of the particular song samples we isolated. In the end, we created one .txt file, bearing the name of each corresponding sample that contained its lyrics.

3.3.4 EEG Data

Theory behind Electroencephalography

Electroencephalography (EEG) is the recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain [41]. If we place an array of electrodes on the scalp, the difference in voltage (caused by the synchronous activity of thousands or millions of neurons that have similar spatial orientation) between any two electrodes can be measured by a voltmeter. Recording these voltages over time gives us the EEG[42].

An internationally recognized method to apply the location of (and also to describe the notation of) scalp electrodes in the context of an EEG test or experiment is the '10-20 system' or 'International 10-20 system'.

Scalp EEG activity shows oscillations at a variety of frequencies. Several of these oscillations have characteristic frequency ranges, spatial distributions and are associated with different states of brain functioning. These are called brain-waves or brain rhythms and while not being definite, they are generally agreed upon by scientists to have the following names and frequency ranges:

Brainwave Type	Frequency Range	Associated Mental States and Conditions
Delta	0.1Hz to 3Hz	Deep, dreamless sleep, non-REM[17] sleep, unconscious
Theta	4Hz to 7Hz	Intuitive, creative, recall, fantasy, imaginary, dream
Alpha	8Hz to 12Hz	Relaxed but not drowsy, tranquil, conscious
Low Beta	12Hz to 15Hz	Formerly SMR[18], relaxed yet focused, integrated
Midrange Beta	16Hz to 20Hz	Thinking, aware of self & surroundings
High Beta	21Hz to 30Hz	Alertness, agitation
Gamma	30Hz to 100Hz	Motor Functions, higher mental activity

Table 3.1: The various brain-waves, their corresponding frequency ranges according to general consensus and the activities or conditions they are associated with.(Table adapted from: http://www.neurosky.com/Documents/Document.pdf?DocumentID=77eee738-c25c-4d63-b278-1035cfa1de92)

17.Rapid eye movement sleep (REM sleep) is a normal stage of sleep characterized by the random movement of the eyes.

18.The Sensory Motor Rhythm (SMR) is a brainwave type that SMR typically decreases in amplitude when the corresponding sensory or motor areas are activated.

However sometimes researchers further divide these ranges into subcategories. In our case, for instance we have:

Name (Symbol)	Frequency Range in Hz
Low alpha	8-9
High Alpha	10-12
Low Beta	13-17
High Beta	18-30
Low Gamma	31-40

Table 3.1: Further subdivisions of the brain-waves and their corresponding frequency ranges according to the model we used (the one used by the Neurosky Mindset).

In addition to their already established medical uses, EEG is used more and more in research, to measure event-related potentials. An event-related potential (ERP) is any electrophysiological response to an internal or external stimulus averaged over a short time. In our case the stimulus is auditory. As the EEG reflects thousands of simultaneously ongoing brain processes, the brain response to a single stimulus or event of interest is not usually visible in the EEG recording of a single trial. Usually many trials are required in order for the measurements to be meaningful. To resolve these potentials against the background of ongoing EEG and other biological signals and ambient noise, signal averaging is usually required. The signal is time-locked to the stimulus and most of the noise occurs randomly, allowing the noise to be averaged out with averaging of repeated responses [43].

Most common sources of noise include:
(i) Biological artifacts. They are electrical signals that originate from non-cerebral origin, but still from the body. Some examples are:

- Eye-induced artifacts (includes eye blinks, eye movements and extra-ocular muscle activity)
- ECG (cardiac) artifacts
- EMG (muscle activation)-induced artifacts
- Glossokinetic artifacts

(ii) Environmental artifacts. They originate from outside the body. They include:

- Movement by the patient
- Settling of the electrodes
- Poor grounding of the EEG electrode
- Interference from external devices

The limitations of the EEG include its poor spatial resolution and its high sensitivity to signals originating from particular areas, especially from the cortex, in expense to neuronal activity originating from other areas. Its main advantage as a brain-activity analyzing technique is that it is simple and non-invasive.

Our EEG data

We used the Neurosky Mindset[19] to obtain the measurements. Neurosky Mindset is a dry (as in 'no conductive material is required') sensor system for consumer applications. It has a single EEG-dedicated sensor placed on a position known as FP1 (in the 10-20 system notation). It is light and soon

19.Neurosky's Official Web Page: http://www.neurosky.com/

the subject forgets they are wearing, allowing for unbiased measurements.

During the annotation of the songs (**Chapter 4**) , one of the annotators was also requested to wear the device and have EEG measurements taken from them. They were given instructions to remain silent, immobile and calm, in order to reduce biological artifacts. The EEG data we used were acquired through this process. They did not undergo some sort of pre-processing.

4. ANNOTATION PROCESS

We explained on the previous chapter how we obtained the data-set of the song samples. Our next step is to have this data-set annotated, so that we will have supervised data with which to train our classifiers. We shall begin by describing the annotation process as a whole.

4.1 Description of the Annotation Process

The main idea was to create a program and distribute it to each of several human annotators. The program would play all the songs of our dataset, in random order, to the annotator. The annotator would then proceed select among the available label values (consult **section 4.2** for more details) the one(s) that they perceived as the most appropriate. The process could be completed in more than one sessions, the annotators could exit the program at any time, and their progress would be saved. In fact they were instructed to complete the annotation in no less than 3-4 sessions, so that they would not become tired and their choices biased.

The annotators could also change their already-placed labels either by specifying the ID number of the song they wish to re-evaluate, or by listening to all the already-labeled songs one after another. They could also skip the annotation of a particular song to a later time, in case of uncertainty. Finally, at any point they could start the annotation from the beginning, losing their progress.

For each annotated song we stored information about the label(s) the annotator selected, the time it was annotated, its relative order of annotation, the session during which it was annotated and, of course, its unique ID which could relate it to its original name.

Once all the songs would be annotated, the annotator would be notified that their task was finished and requested to send us the output file containing the final data (i.e. containing all the information mentioned above). They could still re-evaluate their choices, as changing a label would simply overwrite the final output file to reflect the changes.

Three main versions of the program were implemented due to the use of two distinct labeling schemes (consult **section 4.2**) and the use of two different datasets, derived from two different collections (consult **section 3.3**). The code was written in matlab and the program was distributed to the annotators in CDs or DVDs, depending on its version, along with a readme .pdf file describing its purpose and functions in detail. Instructions were also given to the annotators in person.

Additional instructions given to the annotators included:

- Try to evaluate the intended emotion, not the experienced (consult **section 2.2.1**).
- When uncertain, skip the song and annotate another one.
- When tired, terminate the session and continue at a later time.

4.2 Labeling Methods

Choosing a labeling scheme is vital in order to describe our model and subsequently train it. During our work we experimented with two labeling methods:

- Labeling Method 1: Using MIREX mood clusters, applied to the dataset derived from Collection 1 (Classical Music)

- Labeling Method 2: Using SAMs to describe two emotional dimensions, applied to both datasets

Let us examine them both, discuss their shortcomings and merits and see what changes they entailed to the corresponding versions of program used by the annotators.

4.2.1 Labeling Method 1: Using MIREX mood clusters

Our first approach was to use the MIREX mood clusters we presented in **section 2.2.2**. Since the annotators were all native Greek speakers, we decided to translate the clusters in Greek as follows:

- Cluster_1: παθιασμένη, διεγερτική, δυναμική, παρακινητική, εμψυχωτική
- Cluster_2: πρόσχαρη, χαρωπή, εύθυμη, γλυκιά, διασκεδαστική
- Cluster_3: θλιβερή, μελαγχολική, νοσταλγική, συγκινητική, καταθλιπτική
- Cluster_4: αστεία, παράξενη, πνευματώδης, ειρωνική, παιχνιδιάρικη
- Cluster_5: φλογερή, έντονη, τεταμένη/αγχώδης, αγωνιώδης, εκρηκτική, άστατη

As we can see, instead of translating the 'moods' word by word, we opted for a more 'free' translation that 'captures the general feeling' of each cluster.

The program used a console interface and the annotators could place either one label to each song, or two (a primary one and a secondary one). We can see a sample of the interface in f**ig. 4.1**.

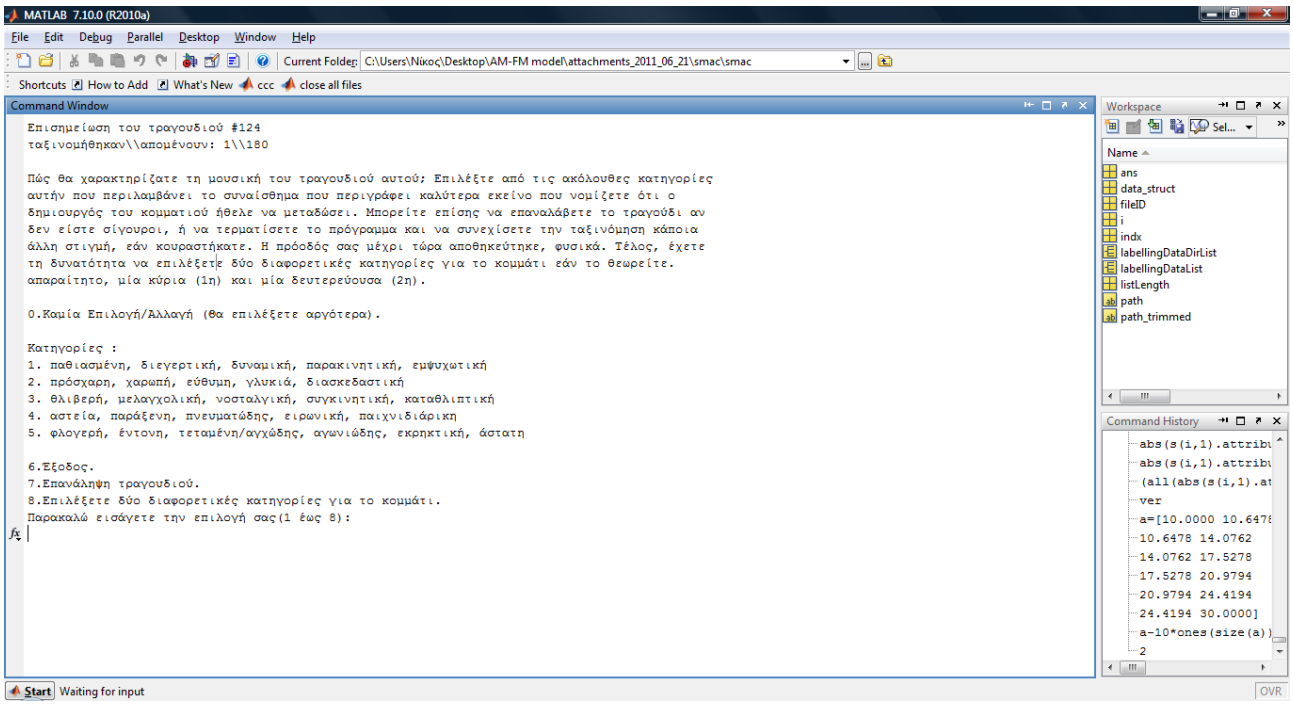


Figure 4.1: An example of the interface of the program used during the annotation (MIREX Mood Clusters Version)

In **section 4.4.1** we will see which measures we used for evaluating the overall agreement scores observed among the annotators when this labeling scheme was utilized. In **section 4.5.1** we will see the methods used for assigning final labels to the samples annotated with this annotation method.

Unfortunately, this method did not prove so successful, and in the end it was dropped in favor of the

one described below. Thus, this method was only used during the annotation of the samples belonging to Collection 1. And since Collection 1 was also abandoned in favor of Collection 2, other than in the aforementioned sections, we will not discuss this method any further.

4.2.2 Labeling Method 2: Using SAMs

Next, we decided to use a two-dimensional space to describe the emotion, consisting of the dimensions of valence and activation, as presented in **section 2.2.3**. The annotators had to select for each dimension one of 5 possible ordinal integer values that best describe emotion in that dimension. This time, we opted for a graphical user interface which was far more user-friendly and made use of the Self-Assessment Manequins (SAMs) proposed by [44] and extensively used in similar applications to represent the dimensional values of a dimensional emotion. In fig. 4.2 we can see the new interface's 'main menu' and in fig. 4.3 we can see its 'annotation screen', as well as the SAMs.

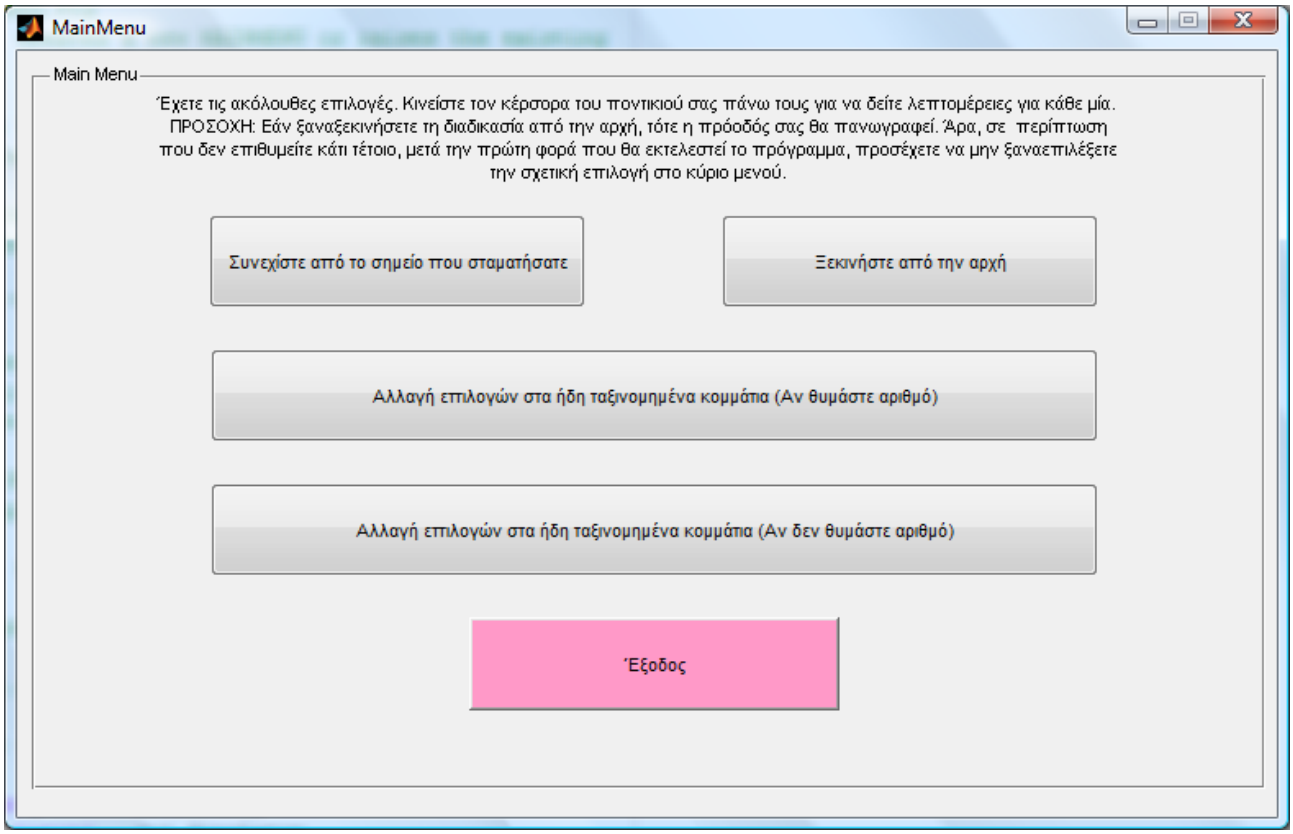


Figure 4.2: The main screen of the program used during the annotation (GUI using SAMs for two-dimensional emotion representation Version)

While a song is playing a taskbar is shown depicting its progress. The interface is locked until the song stops playing and the taskbar vanishes. Now, the user can select the SAM icons they consider appropriate (only one is allowed for each dimension, selecting one while another one is already selected cancels the old one). When a SAM is selected it is marked with a '✓'. When a SAM is selected in each of the two dimensions, the choice to save and move on to the next sample becomes available (fig. 4.4).

Other than these, the program at its core remained unchanged. The final output contains the user's labels in each emotional dimension.

In **section 4.4**, we will see the annotators' personal statistics. In **section 4.4.2** we will see the overall

agreement scores observed among the annotators when this labeling scheme was utilized. In **section 4.5.2** we will see the methods used for assigning final labels to the samples annotated with this annotation method.

As we will observe in all three cases, this method proved to be quite successful, and it was preferred over the one described above. Thus, it was used during the annotation of both sample collections.

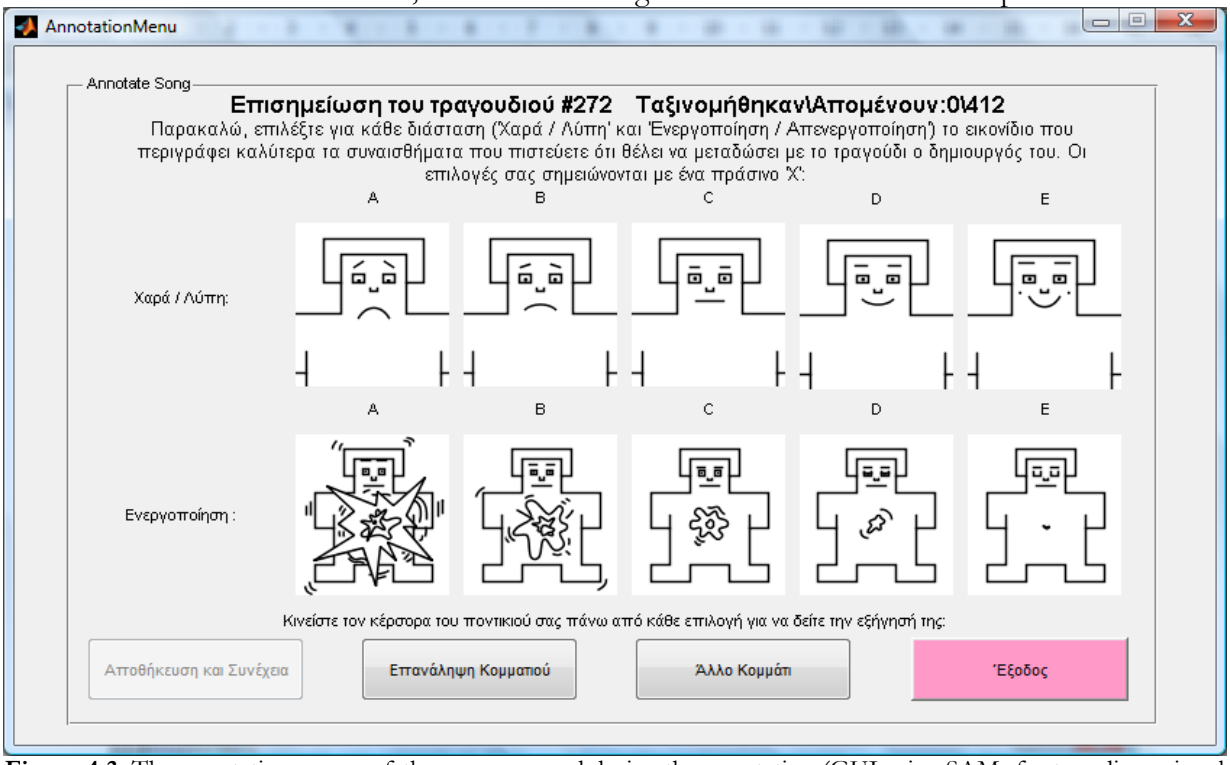


Figure 4.3: The annotation screen of the program used during the annotation (GUI using SAMs for two-dimensional emotion representation Version). We can also see the SAMs for the two dimensions (valence: up/ activation: down). Note that the numeric values that correspond to each SAM have been substituted by letters. Numbers would cause the user to think quantitatively, which is something we wish to avoid.

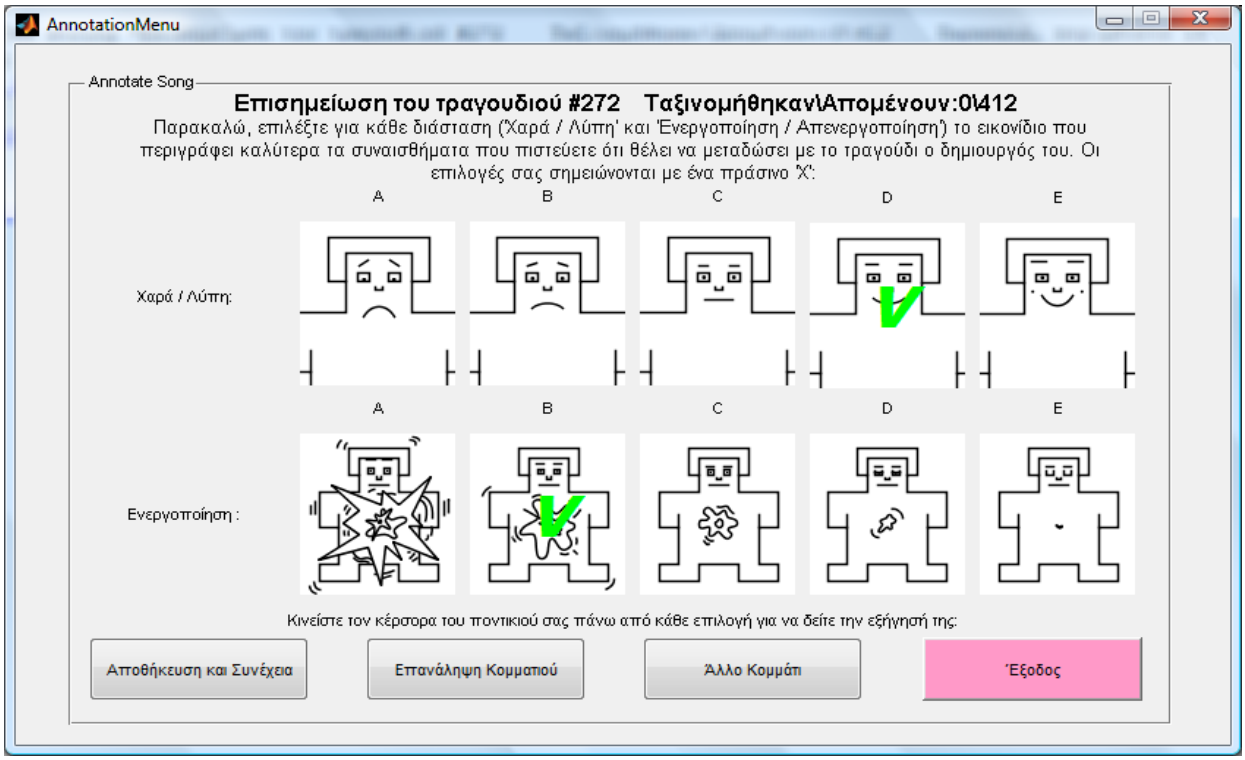


Figure 4.4: Same as the above. Notice the 'tick' symbols and choice 'Save and Continue' that has now become unlocked.

4.3 Annotators' Personal Statistics

After the annotation stage was completed, we calculated some statistical measures and constructs to visualize the distribution of labels, their possible correlations, possible biases of certain annotators, differences between the two dimensions, etc., in an effort to understand our problem a little more deeply. Here we will present some personal statistics of each annotator for the labels obtained using the SAMs method for both datasets. The results were generated by a matlab script we implemented.

For the Dataset No 1 (Classical Music):
First we present the first-order statistics (mean and standard deviation for the individual annotators in each dimension separately (**Table 4.1**).

ANNOTATOR	VALENCE		ACTIVATION	
	Mean	Standard Deviation	Mean	Standard Deviation
1	3.0884	0.89624	3.1050	1.16190
2	2.7514	1.04830	2.4144	1.15930
3	2.9945	1.07750	3.1271	1.38660

Table 4.1: Annotators' Personal Statistics for Dataset 1 (Classical Music Samples) using the Valence-Activation Dimensions.

2-D Histograms of Valence Vs Activation Vs Number of Occurrences: They count the co-occurrences of particular valence (column) and activation (row) values. We present it both in a graphical form light boxes correspond to high number of occurrences, and in numerical form (Activation by Valence matrix with each cell c_{ij} containing the number of samples for which the annotator chose $Valence = j$ and $Activation = i$).

Annotator 1:

Activ. \ Valence	1	2	3	4	5
1	0	1	1	0	0
2	4	10	8	21	8
3	2	15	12	24	12
4	3	24	16	11	1
5	4	3	1	0	0

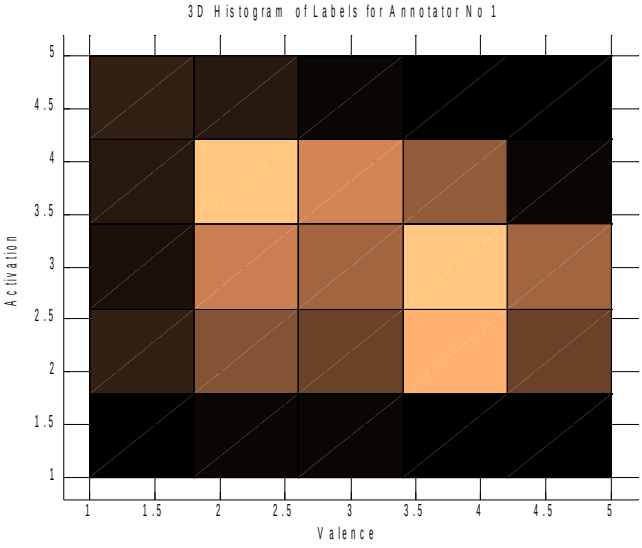


Figure 4.5: Valence-Activation 2-dimensional histogram for Annotator 1. (Attention, in the matrix the activation values are increasing as we move down, while on the figure they are increasing as we move up).

Annotator 2:

		Valence				
		1	**2**	**3**	**4**	**5**
Activ.	**1**	10	3	2	1	0
	2	10	20	22	11	7
	3	12	18	7	9	1
	4	10	11	13	4	1
	5	5	3	1	0	0

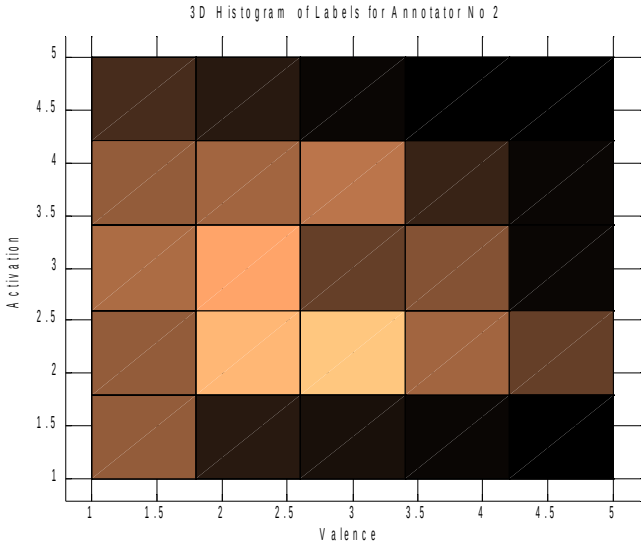


Figure 4.6: Valence-Activation 2-dimensional histogram for Annotator 2. (Attention, in the matrix the activation values are increasing as we move down, while on the figure they are increasing as we move up).

Annotator 3:

		Valence				
		1	**2**	**3**	**4**	**5**
Activ.	**1**	5	1	1	3	4
	2	4	9	14	13	8
	3	5	13	9	10	22
	4	6	14	9	8	8
	5	5	9	0	1	0

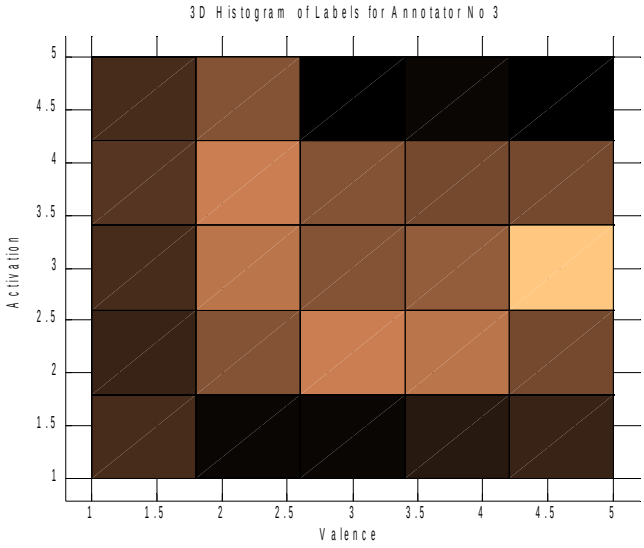


Figure 4.7: Valence-Activation 2-dimensional histogram for Annotator 2. (Attention, in the matrix the valence values are increasing as we move down, while on the figure they are increasing as we move up).

Comments:

- Let us take a look at **Table 4.1**. We notice that the distributions of labels of Annotator 1 and Annotator 3 are centered around the median value of 3 in the Valence dimension, and the value 3.1 (still close enough to the median, but also indicating a small bias towards low[20] Activation) in the Activation dimension. Annotator 2 seems to have a very strong bias towards negative Valence and high Activation. Finally, in the Activation dimension's distribution we observe larger values of standard deviation from the mean than in Valence.
- Examining **figures 4.5 - 4.7**, we can also notice that Annotator 2 is biased towards low numeric

20.Remember the SAMs: Activation begins with 'High Activation' (1) and moves towards 'Low Activation' (5)

values in both dimensions (**fig. 4.6**). As for the other two annotators, Annotator 1 seems to correlate high absolute values of valence with high activation (**fig. 4.5**), as is usually the case in such studies [26], [29], [45]. In fact, Annotator 1 is a musician so their assessment has some added merit. Annotator 3, on the other hand, demonstrates quite the opposite behavior, and also, displays a strong tendency towards the pair $(Activation, Valence)=(3,5)$ for some reason.

For Dataset No 2 (The Beatles' Songs):
Let it be noted that Annotators 1 and 2 are the same persons that participated in the annotation of Dataset 1, as well. Annotator 3 is a different person in each case. The first-order statistics of the label distribution for each annotator per dimension are (**Table 4.2**):

ANNOTATOR	VALENCE		ACTIVATION	
	Mean	Standard Deviation	Mean	Standard Deviation
1	3.3447	0.89206	3.2015	0.91524
2	3.2524	1.05300	2.6529	1.01730
3	3.0316	0.87210	3.2646	0.92828

Table 4.2 Annotators' Personal Statistics for Dataset 2 (Beatles Music Samples) using the Valence-Activation Dimensions.

2-D Histograms of Valence Vs Activation Vs Number of Occurrences: They count the co-occurrences of particular valence (column) and activation (row) values. We present it both in a graphical form light boxes correspond to high number of occurrences, and in numerical form (Activation by Valence matrix with each cell c_{ij} containing the number of samples for which the annotator chose $Valence=j$ and $Activation=i$).

Annotator 1:

Activ. \ Valence	1	2	3	4	5
1	0	0	2	3	0
2	0	4	13	59	1
3	3	17	27	68	11
4	9	46	83	40	1
5	1	20	4	0	0

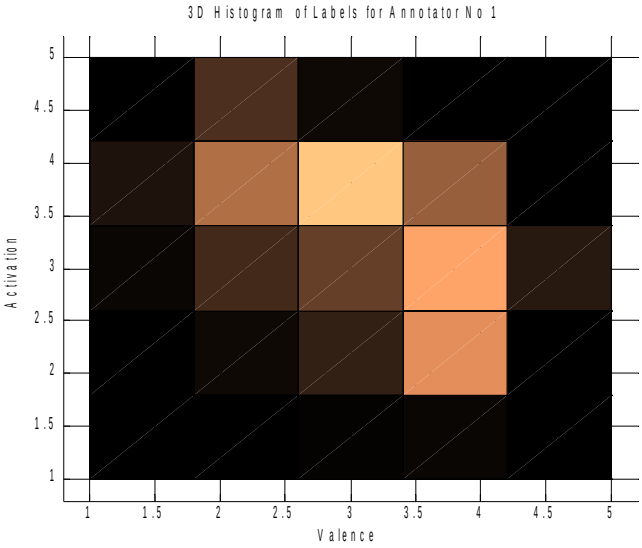


Figure 4.8: Valence-Activation 2-dimensional histogram for Annotator 1. (Attention, in the matrix the activation values are increasing as we move down, while on the figure they are increasing as we move up).

Annotator 2:

		Valence				
		1	**2**	**3**	**4**	**5**
Activ.	**1**	3	6	5	5	4
	2	0	16	32	33	7
	3	8	30	43	11	3
	4	23	71	62	16	2
	5	19	11	2	0	0

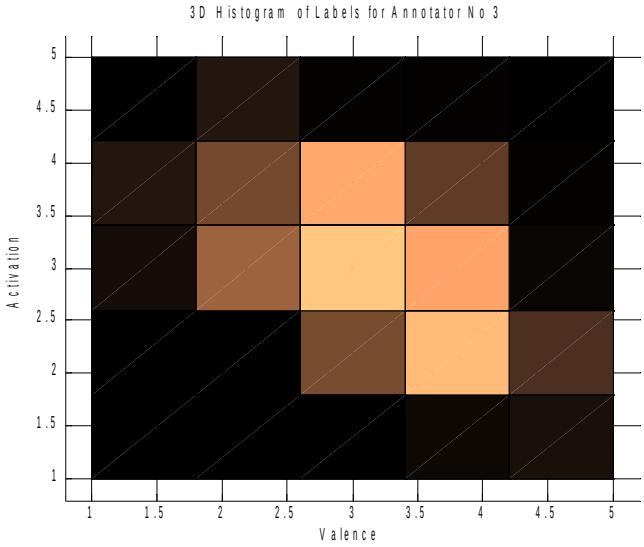


Figure 4.9: Valence-Activation 2-dimensional histogram for Annotator 2. (Attention, in the matrix the activation values are increasing as we move down, while on the figure they are increasing as we move up).

Annotator 3:

		Valence				
		1	**2**	**3**	**4**	**5**
Activ.	**1**	0	0	0	4	6
	2	0	0	26	64	17
	3	5	34	69	57	3
	4	8	25	58	21	2
	5	1	8	2	2	0

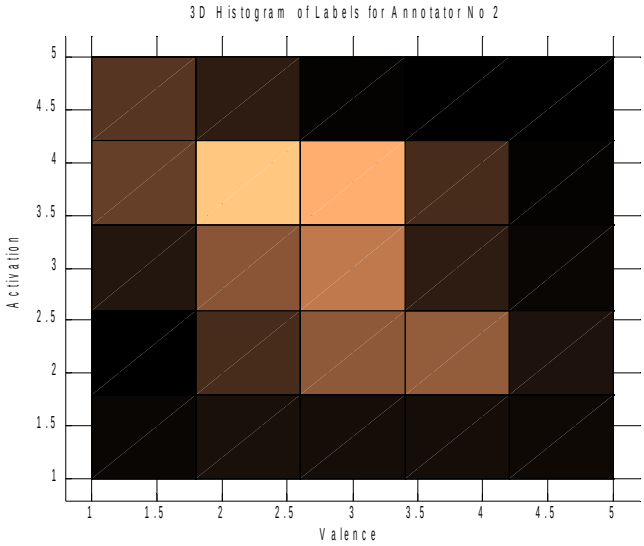


Figure 4.10: Valence-Activation 2-dimensional histogram for Annotator 2. (Attention, in the matrix the valence values are increasing as we move down, while on the figure they are increasing as we move up).

Comments:

- Let us take a look at **Table 4.2**. We notice that the distribution of labels of Annotator 1 and Annotator 2 are centered around a value greater than the median value of 3 in the Valence dimension. Also, the distribution of labels of Annotator 1 and Annotator 2 are centered around a value greater than the median value of 3 in the Activation dimension, as well. Annotator 2 seems to have a very strong bias towards high Activation, while Annotator 3 appears less biased towards positive Valence than the other two.
- Comparing the results of **Table 4.2** to those of **Table 4.1**, we reach the conclusion that The Beatles' song samples are considered by the annotators to be both 'happier' and 'more

activating' than the Classical Music songs we used.

- Examining **figures 4.8 - 4.10**, we can also notice the individual differences among annotators we mentioned above, as well as the general tendency towards positive Valence and high Activation. However, perhaps the most important observation, is that -especially when compared to the corresponding figures of the Dataset 1 (**figures 4.5 - 4.7**), this time they look much similar to one another (all three showing the same tendency towards positive Valence and high Activation).

This last observation implies a greater agreement (see **section 4.4**) among annotators. Combined with the greater number of samples in this dataset, and the ability it granted us to explore other modalities as well, The Beatles' dataset was the one we ended up using for the rest of this study.

In fact, the two dimensions seem somewhat correlated, as high Activation values (low numerical ones) seem to coincide with high Valence values (high numerical ones), while low Activation values (high numerical ones) seem to coincide with low Valence values (low numerical ones).

4.4 Annotators' Agreement

A high agreement among the annotators is a good indicator that our labels are in fact good examples to use for training a model. Let us see the techniques we used in order to evaluate the agreement among annotators along with some examples of Annotator-to-Annotator, and Overall Agreement results. All were generated by matlab scripts we implemented in matlab.

4.4.1 Agreement Evaluation Techniques for Labeling Method 1 (Mood Clusters)

The results were not that good and the method was abandoned. We just mention the methods for completeness' sake.

Each annotator chose one primary and (optionally) one secondary label. In order to quantify their agreement, we used three criteria:

- 'Strictest': Considering only exact matches, (i.e. Both the primary and the secondary labels are the same) as matches
- 'Least Strict': Considering semi-matches (one out of two labels match) as matches as well
- 'Exact': Considering semi-matches as 0.5 of a match each.

4.4.2 Agreement Evaluation Techniques for Labeling Method 2 (Valence-Activation)

The annotators chose two labels each, one for each dimension (Valence, Activation). We treated the labels as ordinal values, that is, not as simple labels marking a distinct category, but as points on an dimensional scale, as they were intended to be used from the beginning. In other words a Valence of 1 is smaller than a Valence of 2 and a difference between Activation 2 and Activation 5 is greater than that of Activation 4 and Activation 5.

First we calculated a number of statistics related to the absolute distances between each pair of annotators' corresponding labels per dimension, including histograms of occurrences, and a rough estimation of agreement, both per dimension, but also an 'overall' one, averaged over both dimensions, using just these differences.

Then we calculated Krippendorff's alpha [46] for ordinal data, in each dimension, which is a measure of agreement that takes into account the expected agreement as well (agreeing upon a label does not

necessarily mean that the label is 'reliable', it could be just a random agreement). The Krippendorff's alpha was calculated as follows:

$$alpha = 1 - \frac{Observed\ Disagreement}{Expected\ Disagreement}$$

Where:

Observed Disagreement is the average over all disagreements (measured as absolute differences) in the data and

Expected Disagreement is the average difference between any two values i and k over all $n(n-1)$ pairs of values possible within the data . It is the disagreement that is expected when the values used by all annotators are randomly assigned to the samples.

Table 4.3 and **Table 4.3** show the Observed Agreement based on the following formula for each dimension:

$$1 - E\{\frac{|(l_{ij} - l_{ik})|}{4}\}$$

Where l_{ij} is the label Annotator j assigned to sample i and l_{ik} is the label Annotator k assigned to sample i. They are normalized by 4, because since all annotators must assign one of the labels $\{1,2,3,4,5\}$, the greatest absolute difference possible between l_{ij} and l_{ik} is 4.

Now let us see the agreement results we obtained from the two datasets we used.

For Dataset No 1 (Classical Music):

First, on **table 4.3**, we see the observed disagreements among all pairs of annotators:

Annotators	**Observed Agreement**		
	Valence	**Activation**	**Overall**
1, 2	**0.79144**	**0.77486**	**0.78315**
1, 3	**0.81906**	**0.81492**	**0.81699**
2, 3	**0.83149**	**0.75552**	**0.79351**

Table 4.3: Agreement Statistics between all annotator pairs in each dimension, for all song samples.

The histograms of the absolute differences of l_{ij} and l_{ik} are also a good indication of the high agreement among the annotators, as large differences tend to be very rare:

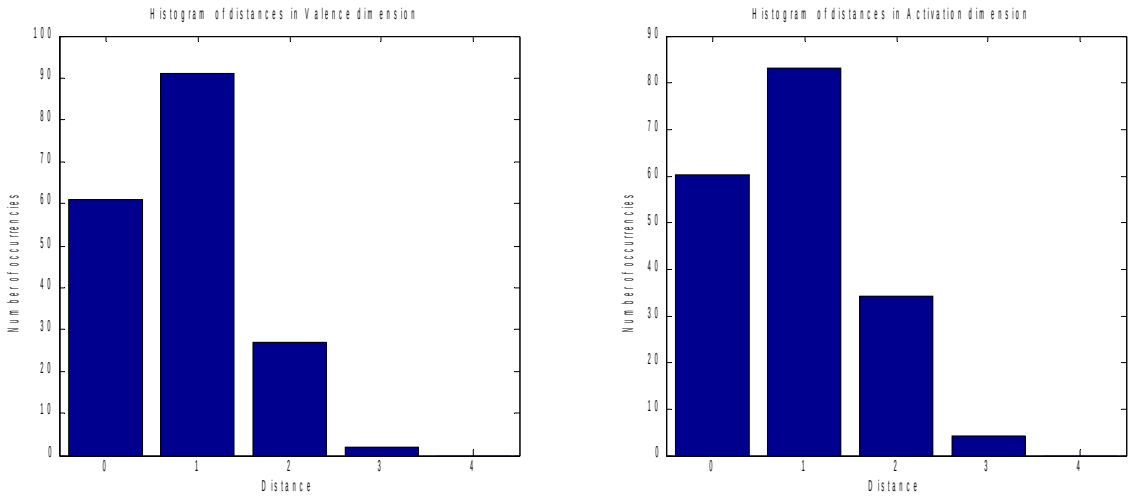


Figure 4.11: Histograms of absolute differences between the labels of Annotators 1 and 2 in the dimensions of Valence (left) and Activation (right)

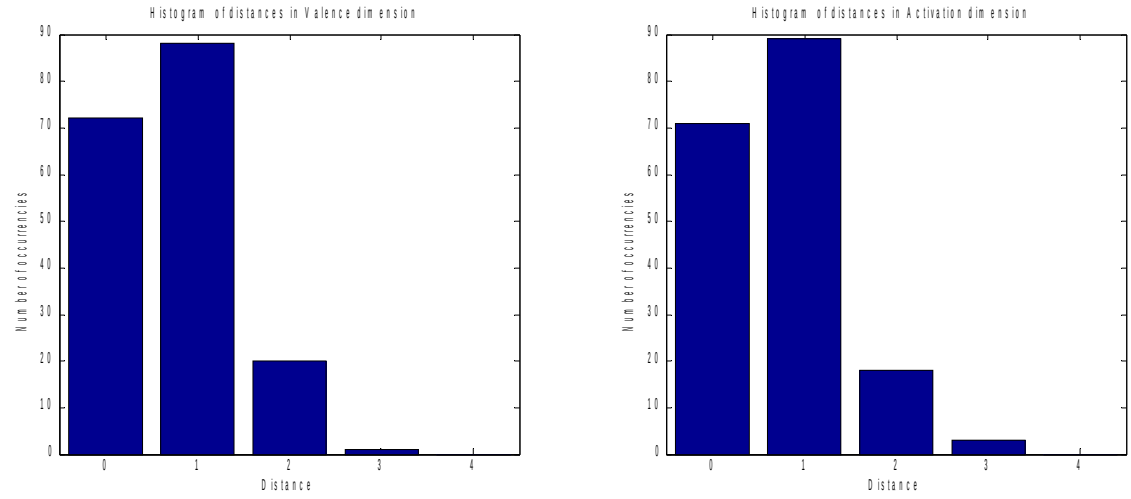


Figure 4.12: Histograms of absolute differences between the labels of Annotators 1 and 3 in the dimensions of Valence (left) and Activation (right)

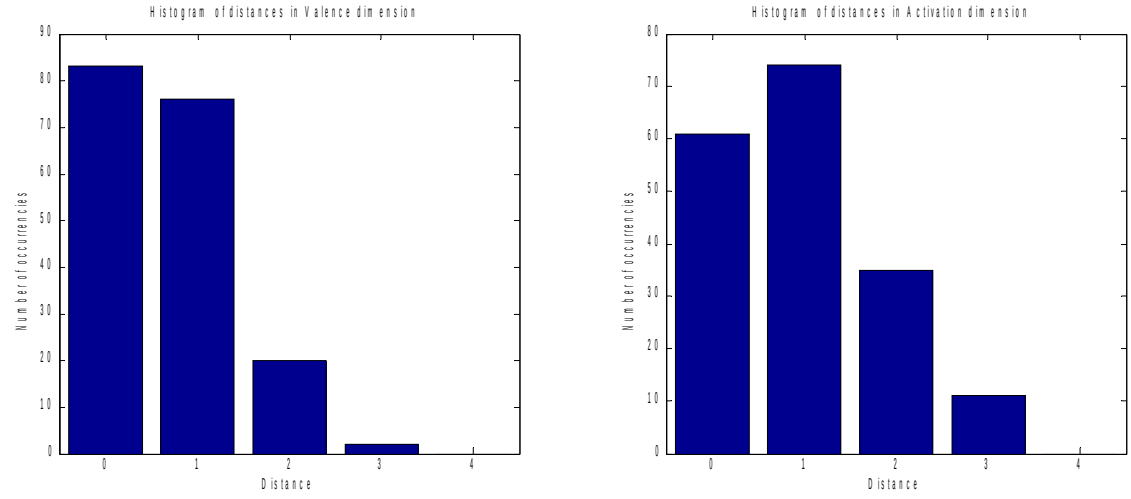


Figure 4.13: Histograms of absolute differences between the labels of Annotators 2 and 3 in the dimensions of Valence (left) and Activation (right)

Krippendorff's alpha (among all annotators)	
Valence	**Activation**
0.4885	**0.5759**

Table 4.4: Overall Krippendorff's alphas per dimension.

Comments:
From all the measures and figures above, it is apparent that the agreement among annotators is quite high, something not trivial on a task that has to do with emotion. Still, as we can see from the histograms (4.11-4.13), two annotators having assigned the same label is less frequent than having assigned two neighboring labels. Also, differences of 4 are non-existent and differences of 3, though infrequent, arise more often in the dimension of Activation. Such a finding is to be expected. Valence is perceived as a bipolar dimension with negative values (sadness) and positive (happiness). A difference of 3, implies that the one annotator perceives the song as 'happy' and the other as 'sad', a very uncommon thing. On the other hand, Activation does not have this trait, so more differences of 3 are encountered.

For Dataset No 2 (The Beatles' Songs):
On the second dataset we obtained similar results:

Annotators	**Valence**		**Activation**		**Overall**
	Observed Agreement	**Krippendorff's alpha**	**Observed Agreement**	**Krippendorff's alpha**	**Observed Agreement**
1, 2	**0.82282**	**0.4930**	**0.79612**	**0.3813**	**0.80947**
1, 3	**0.81978**	**0.4081**	**0.84587**	**0.5028**	**0.83283**
2, 3	**0.83149**	**0.5167**	**0.75552**	**0.4029**	**0.79351**

Table 4.5: Agreement Statistics between all annotator pairs in each dimension, for all song samples.

The histograms of the absolute differences of l_{ij} and l_{ik} are also a good indication of the high agreement among the annotators, as large differences tend to be very rare:

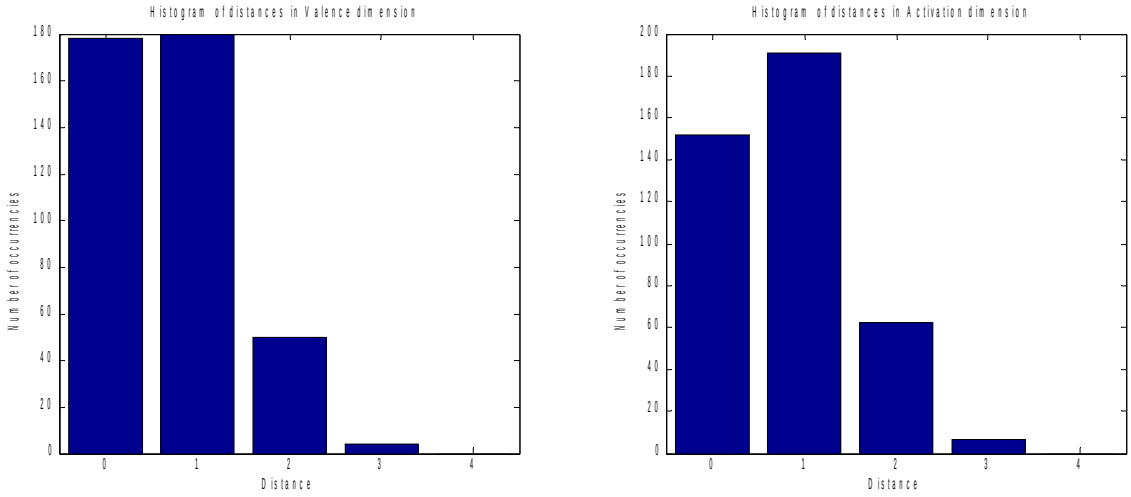


Figure 4.14 Histograms of absolute differences between the labels of Annotators 1 and 2 in the dimensions of Valence (left) and Activation (right)

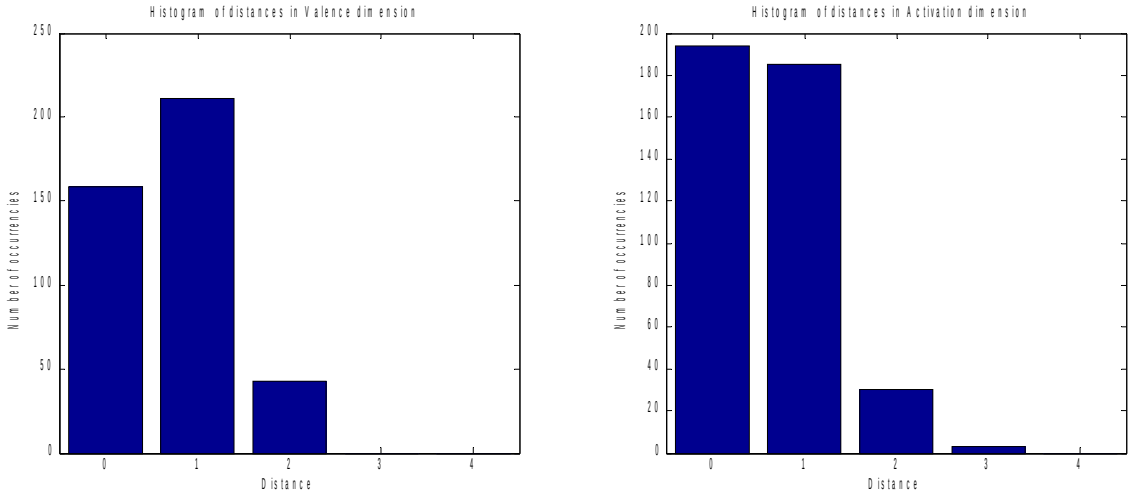


Figure 4.15: Histograms of absolute differences between the labels of Annotators 1 and 3 in the dimensions of Valence (left) and Activation (right)

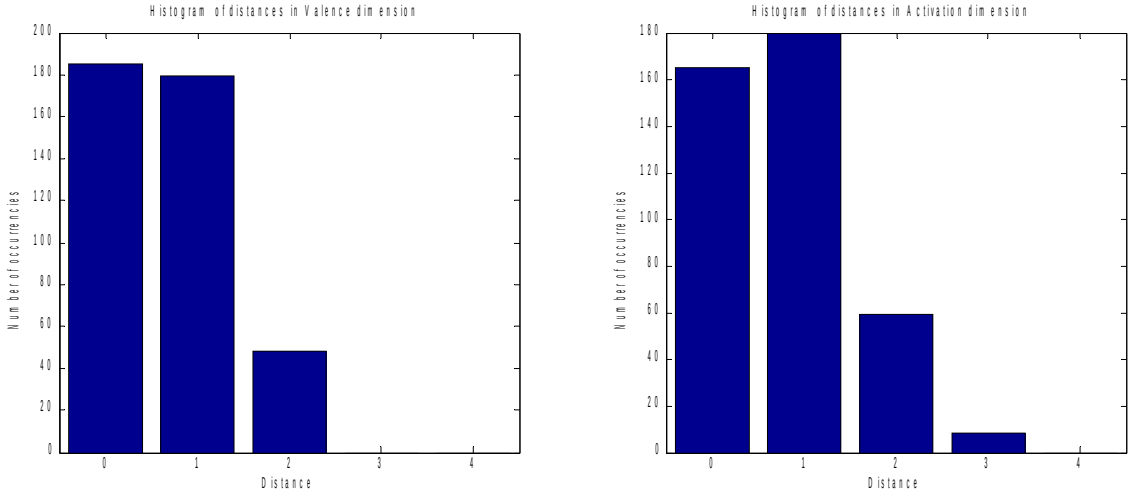


Figure 4.16: Histograms of absolute differences between the labels of Annotators 2 and 3 in the dimensions of Valence (left) and Activation (right)

Krippendorff's alpha (among all annotators)	
Valence	**Activation**
0.4747	**0.4398**

Table 4.6: Overall Krippendorff's alphas per dimension.

Comments:
Nothing important changed with this dataset. The agreement is very high among the annotators.

4.5 Assignment of Final Labels

We are almost ready to move on to the feature extraction and annotation stages. The last thing remaining is to combine the labels of all the annotators into one.

4.5.1 Assignment of Final Labels for Labeling Method 1

The results were not that good and the method was abandoned. We just mention the methods for completeness' sake.

We treated the mood cluster labels as nominal, that is a 1 and a 2 are no more related than a 1 and a three, they are just the 'names' of their corresponding categories.

In this case, we needed a 'strong majority' of the annotators to have the same label assigned to a sample in order to choose it for later use and have that label placed on it. The agreement was not high enough, we relaxed more and more the 'strong majority' criterion, to the point of it becoming plain 'majority'. In the end we turned to method 2 (Valence-Activation)

Reasons for abandoning the 'MIREX mood clusters' method:

- The annotators seemed to favor some categories (2, 3) and ignore others almost entirely (4). The classification results were extremely poor due to the classifiers being unable to generalize over the misrepresented classes. Also, categories 2 and 3 roughly correspond to 'happy' and 'sad', respectively. So we opted to drop the many classes in favor of a model containing only two, actually being the dimension of valence. In the end, we chose a more straightforward way to do this, by using the Valence-Activation model.
- The secondary 'label' choice was rarely used maybe due to practical reasons brought upon by the use of the console interface. It proved out to only complicate things, so in the end it came to be ignored.
- We observed, by looking at the confusion matrices among annotators that classes 2 and 3 (roughly 'happy' and 'sad') were confused with very high frequency. This was unacceptable, as a good model should -at least- discriminate between the two.
- The agreement statistics were very bad among individual annotators.
- When we decided to use only samples for which the agreement was high, we came up with an extremely small number of samples, unable to constitute a meaningful training set. With 8 total annotators, more than 50% agreement among them was observed on the primary labels of only 51 songs.

4.5.2 Assignment of Final Labels for Labeling Method 2

We simply averaged the label every annotator assigned to each dimension for each song. The final labels were no longer integers, but still they were discrete. The values they could take are:

$$\{1, 1.3333, 1.6667, 2, 2.3333, 2.6667, 3, 3.3333, 3.6667, 4, 4.3333, 4.6667, 5\}$$

More on how we treated these labels at the later stages of the classification step will be discussed in **section 7.3**. For the time being, we will treat the samples bearing labels with values below 3 as belonging to one class ('High Activation' or 'Negative Valence', depending on the dimension), the ones bearing labels with values greater than 3 as belonging to the other class ('Low Activation' or 'Positive Valence', depending on the dimension), and the ones with labels of exactly 3 will be ignored.

Finally, let us examine the 2-D Histograms (Valence Vs Activation Vs Number of Occurrences) for the final labels of each of the two datasets examined (**fig. 4.17**):

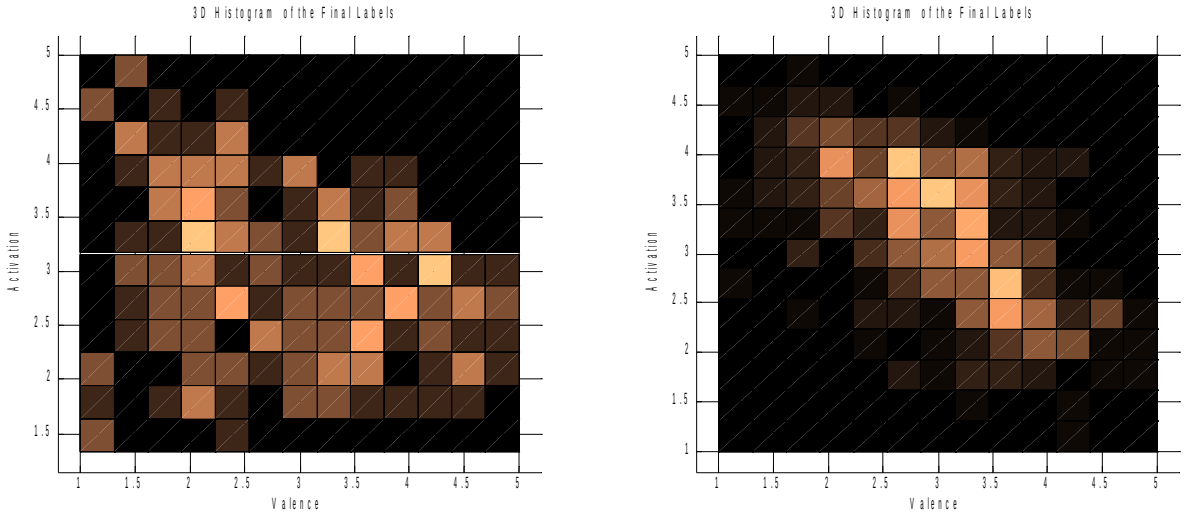


Figure 4.17: The 2-D Histograms (Valence Vs Activation Vs Number of Occurrences) for the final labels of Dataset 1 (left) and Dataset 2 (right). We can still see that the general tendency of the individual annotators towards positive Valence and high Activation can still be discerned on the histogram of their averages. So does the correlation between Valence and Activation , as high Activation values (low numerical ones) seem to coincide with high Valence values (high numerical ones), while low Activation values (high numerical ones) seem to coincide with low Valence values (low numerical ones).
(Remember: high activation corresponds to low numerical values in our scale).

5. FEATURE EXTRACTION PROCESS

As we saw in **section 2.3**, in order to perform classification, we need some features, based on the values of which we can train a model to discriminate between 2 or more classes (during training) and (during the classification itself) decide to which class the item to be classified belongs.

During our work we experimented with features derived from multiple modalities, including the sound signal itself (**section 5.1**), the chords of the song (**section 5.2**), an EEG scan performed on an annotator during the annotation stage (**section 5.3**) and the lyrics of the song (**section 5.4**). As a special category of features extracted from the sound signal, we also studied a group of features derived from the AM-FM (Amplitude Modulation – Frequency Modulation) modeling of the signal, which we will study separately in **section 5.4**, as it represents a different way of treating the signal with its own theoretical background.

Let it be clarified for the last time that the dataset we finally utilized is Dataset 2 (Beatles' song samples).

5.1 Sound Signal Features

All these features, with the exception of the Mel-frequency cepstral coefficients (MFCCs), were extracted with the use of MIRtoolbox [37]. The processing of the samples was performed on a frame basis, that is: as short-term features. Then a number of statistics were calculated from the distribution of the feature values extracted for each frame (mean, standard deviation, maximum value, minimum value, percentiles of the distribution). This was done for two main reasons:

- To reduce the feature space to a smaller one, thus minimizing the effect of the 'curse of dimensionality'[21]
- To obtain more meaningful features, as we study the average emotion of the entire sample, fluctuations on a frame basis are not that important.

The default frame size used by the MIRtoolbox's extractors is 0.05 seconds. The default option concerning the distance of these frames is for them to be half overlapping (each frame begins at the middle of the previous one). Unless specified otherwise, this will be the frame decomposition of the signal used for the calculation of the features.

Let us examine each feature individually.

5.1.1 Rhythmic Periodicity Along Auditory Channels (Fluctuation)

One way of estimating the rhythmic is based on spectrogram computation transformed by auditory modeling and then a spectrum estimation in each band [47].

The steps of the process as executed by Mirtoolbox's function `mirfluctuation` are the following:

(i)The spectrogram (effectively a time vs frequency array with its elements being the corresponding energy intensity values) of the song sample is computed on half overlapping frames of 23 ms. First the Fast Fourier Transform (FFT) of the frame is computed and then the energy in each critical band is

21.The 'curse of dimensionality' refers to the fact that some problems become intractable as the number of the variables increases. In other words, their complexity grows.

calculated.

(ii)Then the Terhardt outer ear modeling[48] is computed, with Bark-band redistribution of the energy, and estimation of the masking effects [49]. The result is expressed in decibels (dB).

The Terhardt outer ear model is shown in **fig. 5.1**, along with a flatter modified version of it proposed by Pampalk.

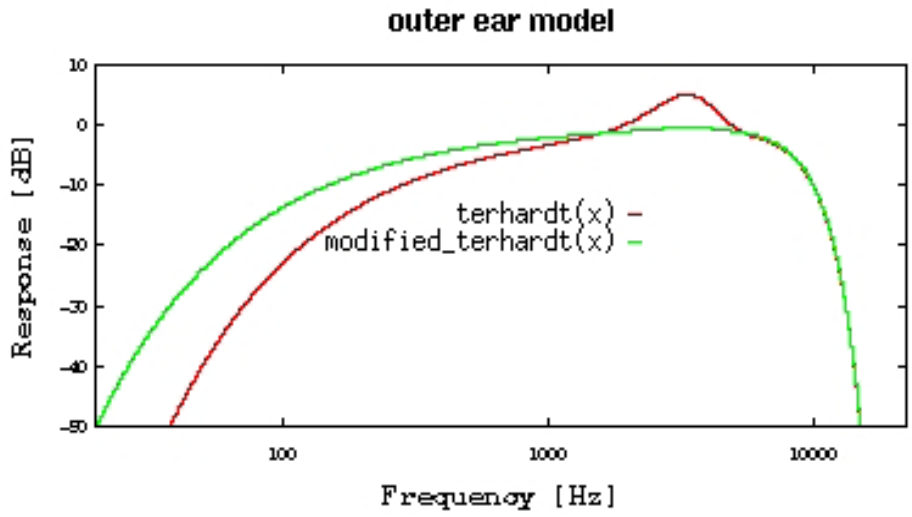


Figure 5.1: The Terhardt Outer Ear Model (red), along with a modified version of it proposed by Pampalk (green). (Image From: Eduard Aylón i Pla, *Automatic detection and classification of drum kit sounds*, MA Thesis, 2006)

The simulation of the outer ear frequency response[22] by a frequency weigthing function proposed by Terhardt[48] expressed in decibels is as follows:

$$A_{dB}(f_{kHz}) = -3.64 f^{-0.8} + 6.5 e^{-0.6(f-3.3)^2} - 10^{-3} f^4$$

As we can see in the above figure, this function (red curve) gives a remarkable emphasis to frequencies around 4kHz (due to the high weight of the exponential part). Alternatively, in order to obtain a flatter response, the following expression was proposed by Pampalk (green curve) :

$$A_{dB}(f_{kHz}) = -3.64 \times 0.6 f^{-0.8} + 0.5 e^{-0.6(f-3.3)^2} - 10^{-3} f^4$$

(the exponential part's weight has been decreased)

We saw in section 2.1.2 what the Bark scale is. The human ear has evolved in such a way that it is able to detect certain frequencies very successfully, while others not. Especially in the range of the human voice we are very capable of hearing well and distinguishing between frequencies. However, on the higher frequency ranges (above 11500 Hz), humans are unable to so. This holds true for the very small frequency ranges, as well. Therefore, to correctly describe the sound color of a song we need to take into account how well the human ear perceives these frequencies. The Bark Scale is a measure of such a thing, with the various frequency ranges the human ear can distinguish being mapped to the 24 values of the scale.

We remind here that the Bark scale ranges from 1 to 24 and corresponds to the first 24 critical bands of hearing. The subsequent band edges are (in Hz): 20, 100, 200, 300, 400, 510, 630, 770, 920, 1080,

22 The frequency response when the input signal is a pulse.

1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500.

The mapping between frequencies in Hz and Barks is shown in **fig 5.2**:

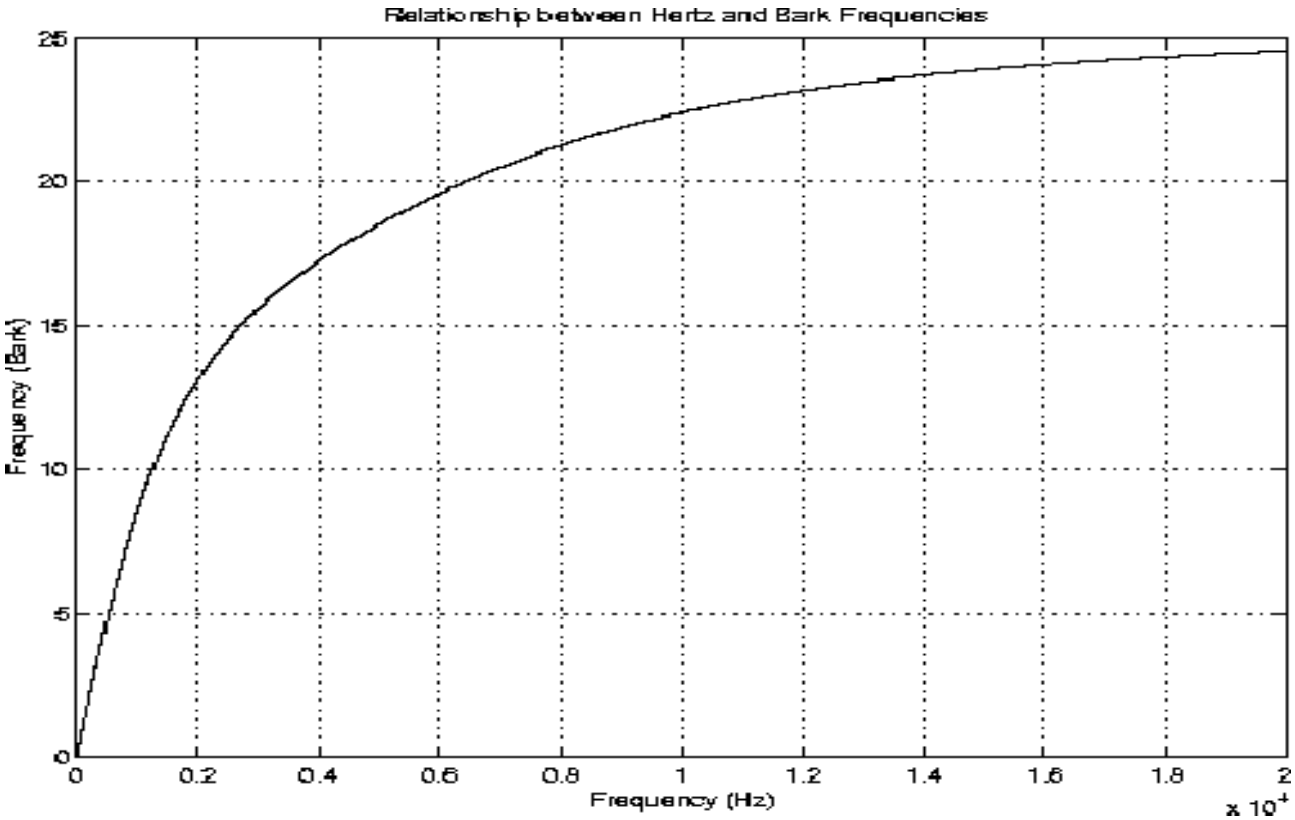


Figure 5.2: The Bark Scale and the corresponding Frequencies in Hz. (Image from: W.A.V.S. Compression Project, http://www.aamusings.com/project-documentation/wavs/psychoAcoustic.html)

To convert a frequency f (Hz) into Bark we use the following formula:

$$Bark = 13\arctan(0.00076\, f) + 3.5\arctan((f/7500)^2)$$

(iii)Then, an FFT is computed on each Bark band, from 0 to 10 Hz. The amplitude modulation coefficients are weighted based on the psychoacoustic model of the fluctuation strength [50]. In the resulting matrix, we can see the rhythmic periodicities for each different Bark band.

(iv)Finally, we sum the resulting spectrum across all bands, leading to a spectrum summary, showing the global distribution of rhythmic periodicities.

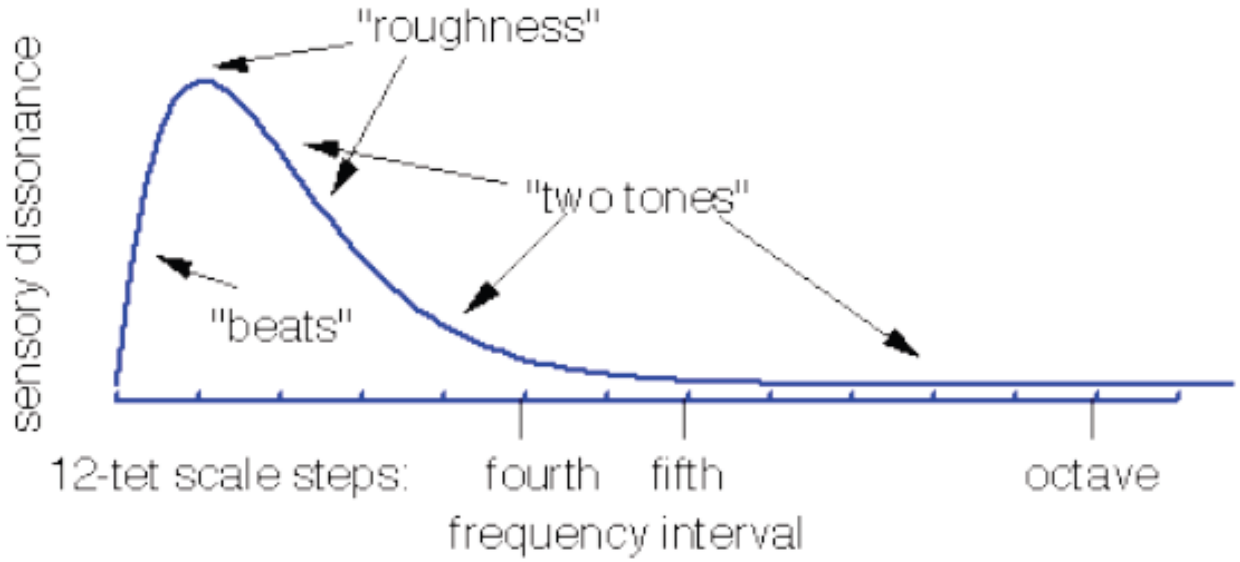
Statistics of the feature we use:
We use as features in our classification the maximum value of the summarized fluctuation, as well as the mean value of the summarized fluctuation.

5.1.2 (Auditory) Roughness

Roughness is an estimation of the sensory dissonance of a sound, first proposed by [51].

According to psychophysical theory, the roughness of a complex sound (a sound comprising many partials or pure tone components) depends on the distance between the partials measured in critical bandwidths.

Any simultaneous pair of partials of about the same amplitude that is less than a critical bandwidth apart produces roughness associated with the inability of the basilar membrane to separate them clearly. Roughness is physiologically determined and therefore universally defined, however, it is appraised differently in different musical styles (some favour it, while others tend to aviod it).
The model that [51] proposed for the estimation of roughness depending on the frequency ratio of each pair of partials (pure tones) is the following:



An estimation of the total roughness of a sound sample is calculated by computing the peaks of the spectrum, and taking the average of all the dissonance between all possible pairs of peaks [52].

Mirtoolbox's function `mirroughness` calculates the total roughness using the above model but also allows for the option to use a variant of the latter with a more complex weighting proposed by [53]. We calculated the average roughness based on this model.

Statistics of the feature we use:
The mean value of the roughness of all frames of a song sample, as well as the standard deviation of these obsevations (i.e. the values of roughness of all frames of a song sample) were among the features we used.

We tried to further refine this feature using instead of the mean value of the roughness of all frames of a song sample, two other features: (i)t the mean value of all frames of the sample with a roughness value lesser than the median, (ii)the mean value of all frames of the sample with a roughness value greater than the median.

5.1.3 Key Clarity

We presented the musical scales in chapter 2. The Mirtoolbox's function `mirkey` gives us a broad estimation of the 7 notes that constitute the (most probable) prevalent scale, calculated through the maximization of the cross-correlation of the chromagram[23] of the sample with that of templates representing all the possible tonalities based on [54] and [55].

Then their 'clarity' or 'key strength' is calculated, which is in essence the probability associated with a particular key.

Statistics of the feature we use:
We use the mean value of the 'key strength' (among all the frames of the sample) as one of our features.

5.1.4 Modality (or Mode)

We already mentioned the major and minor scales and hinted that there also exist other types of scales than these two. However, once we get past the major and minor scales, all the other eight-note combinations are not technically called scales; they're called 'modes'. There are seven essential modes, each of which can be thought of as starting on a different degree of the major scale. We remain within the relative major scale. The only difference is that we just start on different notes.

Modes are important in the construction of melodies. Creating a melody based on a specific mode, allows one to create a different sound or feel while staying within the notes of a traditional major scale (and -as we mentioned- just starting and stopping in different places). Melodies based around specific modes are called modal melodies[24].

The Mirtoolbox's function `mirmode` gives us an estimation of the modality or mode (i.e. major vs. minor) returned as a numerical value: the more it is higher than 0, the more major the given sample is predicted to be, the more the value is lower than 0, the more minor the sample might be.

Statistics of the feature we use:
We use the mean value of this numerical estimate (among all the frames of the sample) as one of our features.

5.1.5 Spectral Novelty

The spectral novelty curve indicates the temporal locations of significant textural changes in the spectrum of a sound sample [56].

The spectral novelty curve is calculated by performing a convolution along the main diagonal of the similarity matrix using a Gaussian checkerboard kernel.

The similarity matrix is the matrix resulting from the mutual comparison between each possible frame analysis in the spectrum of the frame–decomposed song sample. A Gaussian checkerboard kernel is- as the name suggests- a matrix of the form depicted in **Table 5.1**:

23.The chromagram is a redistribution of the spectrum energy along the different pitches (chromas), or classes of pitches (chroma classes).

1	0	1	...
0	1	0	...
1	0	1	...
0	1	0	...

Table 5.1: An n-by-n (square) matrix, whose successive diagonals have all their elements alternately equal to 1 or 0. A size 2-by-2 checkerboard kernel is used here.

Mirtoolbox calculates the spectral novelty curve by use of the command `mirnovelty`. The command's argument is the (frame-decomposed) spectrum of the sound sample. The default frame decomposition was used. The feature we actually use is, in fact, the mean value of the spectral novelty curve that results from the process described here.

An illustration of the process is shown in **figure 5.2**.

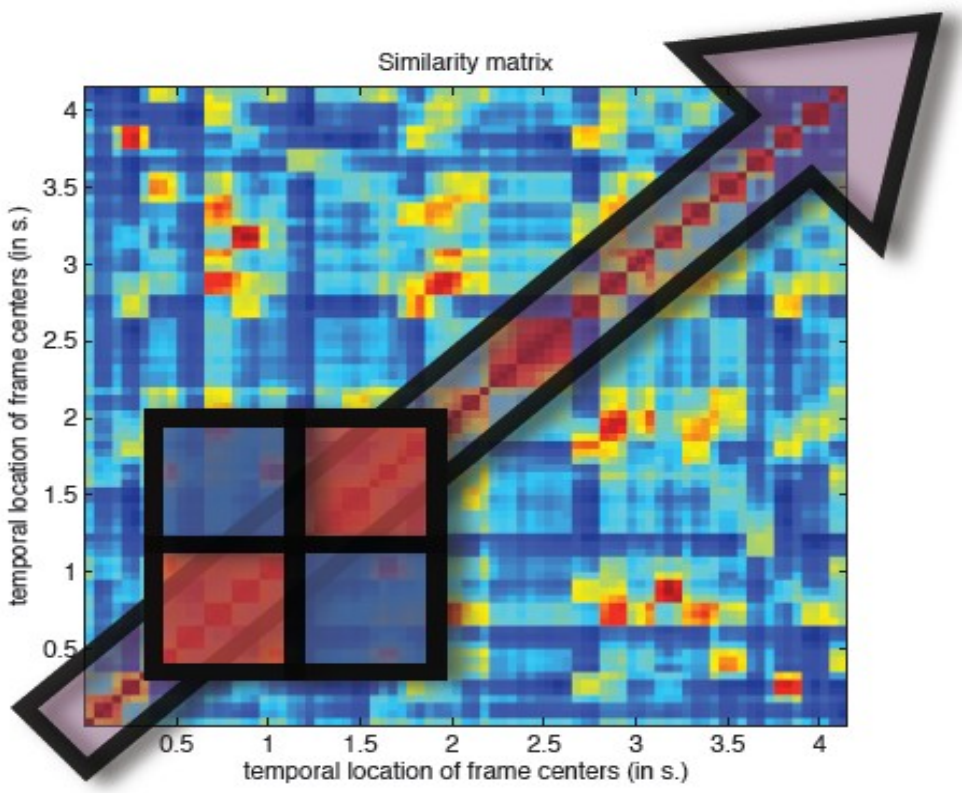


Figure 5.2: An illustration of the process followed to calculate the spectral novelty. (Clarification: the element S_{ij} of the similarity matrix contains information about the spectral similarity between the *i*-th and *j*-th frames. The default distance measure was used, i.e. one minus the cosine of the included angle between observations (treated as vectors)). (Image From: MIRtoolbox 1.3 User's Manual).

5.1.6 Harmonic Change Detection Function (HCDF)

The Harmonic Change Detection Function (HCDF) is the flux of the tonal centroid [57]. In qualitative terms it simply 'tracks the change of the chords'. Algorithmically, it involves the steps shown in **fig. 5.3** [57]:

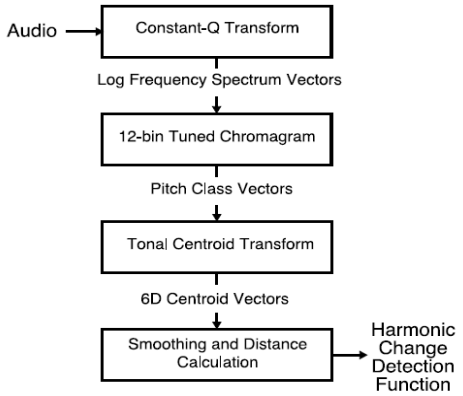


Figure 5.3: An illustration of the process followed to calculate the spectral novelty.

Mirtoolbox calculates the HCDF by use of the command `mirnovelty`.

Statistics of the feature we use:
We use the mean value of the HCDF (among all the frames of the sample) as one of our features.

5.1.7 Mel-frequency cepstral coefficients (MFCCs)

The 'power cepstrum' of a signal is the squared magnitude of the Fourier transform of the logarithm of the squared magnitude of the Fourier transform of a signal [58].

Mel-frequency cepstral coefficients (MFCCs) [59] are coefficients that collectively make up the mel-frequency cepstrum. The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. The mel scale [60] is a perceptual scale of pitches judged by listeners to be equal in distance from one another. In fig. 5.4 we can see how the scale relates to frequencies in Hz.

A popular formula to convert f hertz into m mel is [61] :

$$m = 2595\log_{10}(1+\frac{f}{700}) \quad \textbf{(Eq.5.1.1)}$$

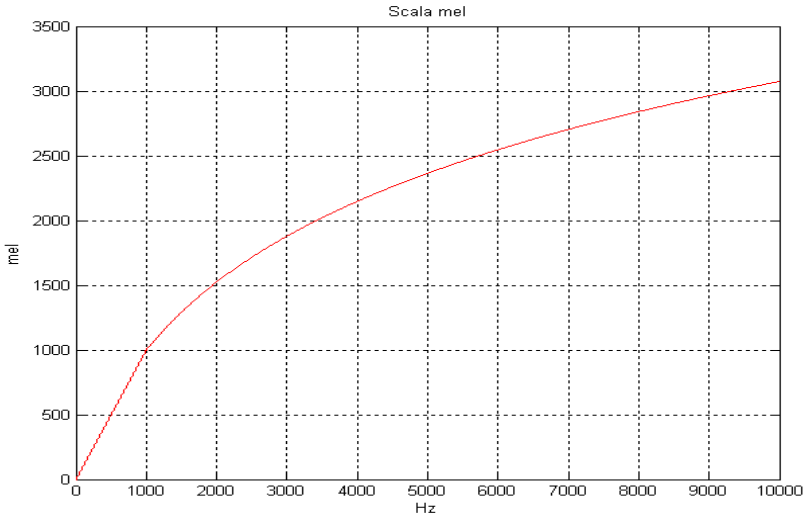


Figure 5.4: The mel scale and the corresponding frequencies in Hz

We will make another reference to the mel scale later, when we will discuss the demodulation of the FM-AM signal, as it is one of the scales used to calculate centers of the filterbanks used for the demodulation.

Now, let us return to the MFCCs. These coefficients are commonly used as features in speech recognition systems. They are also common in speaker recognition, which is the task of recognizing people from their voices.[62]. MFCCs are also increasingly finding uses in music information retrieval applications such as genre classification, audio similarity measures, and others [63]. However, experimental findings have demonstrated that the MFCCs achieve poor emotion classification results, at least for emotional speech classification tasks [64], [65], [66].

In our study, as it we will see at the results section (**Chapter 7**) they prove to be fairly good features, but not as good as more music-inspired ones. When used together with the other features in the final classification results, they slightly improve upon the results obtained by the latter.

MFCCs are commonly derived as follows[67] :

1. Take the Fourier transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows (**fig. 5.5**).
3. Take the logarithms of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal (**Eq. 5.1.2**).
5. The MFCCs are the amplitudes of the resulting spectrum.

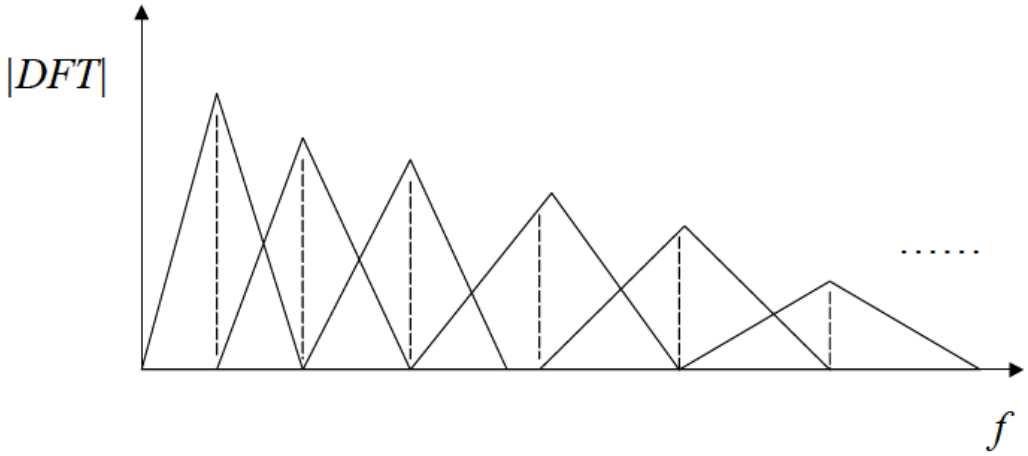


Figure 5.5: Example of triangular filters used to compute MFCCs. (Image From: Jinjin Ye: *Speech Recognition Using Time Domain Features from Phase Space Reconstructions* , MSc thesis, (2004))

The MFCCs are:

$$c[n]=\sum_{m=0}^{M-1} S[m]\cos(\pi n\frac{(m-1/2)}{M}),\ 0\le n<M \quad \textbf{(Eq. 5.1.2)}$$

where *S[m]* is the log-energy at the output of each filter, and *M* is the number of filters, which varies for different implementations from 24 to 40. The advantage of computing MFCCs by using filter energies is that they are more robust to noise and spectral estimation errors [68].

Usually, only the coefficients *c[1]* to *c[12]* are used. . The zeroth coefficient *c[0]* corresponds to the log energy measure (we defined energy in **section 2.1.3**., log energy is just its logarithm).

The features outlined above do not provide temporal information. In order to incorporate the ongoing changes over multiple frames, time derivatives are added to the basic feature vector. The first and second derivatives of the feature are usually called 'Delta coefficients' (or 'Deltas') and 'Delta-Delta coefficients' (or 'Accelerations') respectively. The Delta coefficients are computed via a linear regression formula (Eq. 5.1.3):

$$\Delta c[m]=\frac{\sum_{i=1}^{k} i(c[m+i]-c[m-i])}{2\sum_{i=1}^{k} i^2} \quad \textbf{(Eq. 5.1.3)}$$

where $2k+1$ is the size of the regression window and *c[m]* is the *m*-th MFCC coefficient.

The Delta-Delta coefficients are computed using a linear regression of Delta features.

A typical speech recognition system has a 39-element feature vector. The feature vector consists of 13 static features (12 MFCCs computed from *M* filter banks and log energy), 13 delta coefficients (first derivatives of the static features) and 13 delta-delta coefficients (second derivatives of the static features).

We extracted the MFCCs, along with the MFCC deltas and MFCC accelerations with an existing[24] openSMILE configuration. We used frames of 0.025 seconds taken every 0.010 seconds (0.015 seconds

24.Script written by Florian Eyben, Martin Woellmer, Bjoern Schuller (2009)

overlap between consecutive frames) and worked with 26 bands (*M*=26). We used two methods:

(ι) Subtracting the mean value of the cepstrum from all cepstral features frame by frame
(ιι) Performing no cepstral mean subtraction

The features generated by the 1st method worked slightly better than those generated by the 2nd one.

5.2 Chord Features

Now we will describe in brief the features extracted through the chords' files of the samples. We used matlab scripts for the extraction. More on chords can be found in **section 2.1.4**.

5.2.1 Number of Distinct Chords per Sample Duration

We counted the number of distinct chords of each sample and divided it by the sample's duration. The idea behind this feature is that 'richer' samples (i.e. samples with larger values of this feature) were expected to be sadder than simpler ones (samples with smaller values of this feature).

5.2.2 Number & Duration of Minor Chords per Sample Duration

We counted the number of distinct minor chords of each sample and divided it by the sample's duration.

We also calculated the percentage of the total sample time the minor chords occupied (as a ratio of their summarized duration to the duration of the entire sample).

The idea behind this feature is that samples containing more minor chords (both in terms of multitude and in terms of duration) were expected to be sadder than those containing less.

We first wrote a script to find all the chords contained in all of our samples and save them on a .txt file. Then, after observing the notation used in it to denote minor chords, we constructed a regular expression and searched the individual chords files for matches.

5.2.3 Number & Duration of Major Chords per Sample Duration

We counted the number of distinct major chords of each sample and divided it by the sample's duration.

We also calculated the percentage of the total sample time the major chords occupied (as a ratio of their summarized duration to the duration of the entire sample).

The idea behind this feature is that samples containing more major chords (both in terms of multitude and in terms of duration) were expected to be happier than those containing less.

We first wrote a script to find all the chords contained in all of our samples and save them on a .txt file. Then, after observing the notation used in it to denote major chords, we constructed a regular expression and searched the individual chords files for matches.

5.2.4 Number & Duration of Suspended Chords per Sample Duration

We counted the number of distinct suspended chords of each sample and divided it by the sample's duration.

We also calculated the percentage of the total sample time the suspended chords occupied (as a ratio of their summarized duration to the duration of the entire sample).

The idea behind this feature is that samples containing more suspended chords (both in terms of multitude and in terms of duration) were expected to be creating more tension than those containing less.

We first wrote a script to find all the chords contained in all of our samples and save them on a .txt file. Then, after observing the notation used in it to denote suspended chords, we constructed a regular expression and searched the individual chords files for matches.

5.2.5 Number & Duration of 7th Chords per Sample Duration

We counted the number of distinct dominant 7th chords of each sample and divided it by the sample's duration.

We also calculated the percentage of the total sample time the dominant 7th chords occupied (as a ratio of their summarized duration to the duration of the entire sample).
The idea behind this feature is that samples containing more dominant 7th chords (both in terms of multitude and in terms of duration) were expected to be create more tension than those containing less.

We first wrote a script to find all the chords contained in all of our samples and save them on a .txt file. Then, after observing the notation used in it to denote dominant 7th chords, we constructed a regular expression and searched the individual chords files for matches.

As both dominant 7th chords and suspended chords were expected to create a similar effect, tension, and observing that both were rare (especially suspended chords) in our samples, in the end we merged these 4 features into two:

- Number of Suspended and Dominant 7th Chords per Sample Duration
- Duration of Suspended and Dominant 7th Chords per Sample Duration

5.2.6 Most Probable Key Calculated through Chords

Finally, we attempted an estimation of the song's key (nominal feature) by selecting the most probable of all the key candidates. We based our decision on the chord transposition matrix shown on **Figure 5.6**.

Key Chord Chart

Major Key	I	II	III	IV	V	VI	VII
A	A	Bm	C#m	D	E	F#m	G#dim
B	B	C#m	D#m	E	F#	G#m	A#dim
C	C	Dm	Em	F	G	Am	Bdim
D	D	Em	F#m	G	A	Bm	C#dim
E	E	F#m	G#m	A	B	C#m	D#dim
F	F	Gm	Am	Bb	C	Dm	Edim
G	G	Am	Bm	C	D	Em	F#dim
Minor Key	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**
Am	Am	Bdim	C	Dm	Em	F	G
Bm	Bm	C#dim	D	Em	F#m	G	A
Cm	Cm	Ddim	Eb	Fm	Gm	Ab	Bb
Dm	Dm	Edim	F	Gm	Am	Bb	C
Em	Em	F#dim	G	Am	Bm	C	D
Fm	Fm	Gdim	Ab	Bbm	Cm	Db	Eb
Gm	Gm	Adim	Bb	Cm	Dm	Eb	F

Figure 5.6: A chord transposition table showing the 14 keys we considered as candidate.

For every note we encountered on a sample's chord file, we gave a 'vote' to the key candidates which could contain it. In the end, we chose the key candidate with the majority of 'votes' as the most probable.
In case of ties- and there were many such cases, we were unable to decide and labeled the key as 'Undecidable'.

5.3 EEG Features

Next, we experimented with the features we obtained through the EEG. The data we obtained were measurements every 10msec of seven different brainwaves (more on brainwaves in **section 3.3.4**): Low alpha, High Alpha, Low Beta, High Beta, Low Gamma, Delta, Theta

We first implemented a script in matlab to check for inconsistencies in the data, for example missing files, incomplete measurements etc. Then, we wrote another script that would calculate some statistics of the distribution of the measurements corresponding to an entire song sample. These were:

5.3.1 First Order Statistics of Each Brain Wave

The mean and the standard deviation of the distribution of the measurements of an entire song for each of the seven brainwaves. They were a good indication of the overall activity in each frequency range.

5.3.2 Extrema of Each Brain Wave

The minimum and the maximum values of the distribution of the measurements of an entire song for each of the seven brainwaves. They were an indication of the spikes or valleys activity in each frequency range. However, in a large distribution, these features could be quite random and not hold much of a meaning, so instead of them we ended up using the next two features:

5.3.3 Mean Values of 10% Highest & Lowest of Each Brain Wave

We selected the highest 10% of the values of the distribution and then calculated their mean. We did the same thing for the lowest 10% of the values. Using percentiles instead of single values ensured that the feature would be less prone to noise or randomness

This feature-pack consisted of 4 features per each of the 7 brain waves, therefore 28 total features.

5.4 Lyrics Features

The lyrics files were used as an input to a system that rates the valence of each word, based on a number of 'seed words'. Hit-based metrics among them and the new words estimate the similarity between two words using their frequency of co-existence within larger lexical units (e.g. documents). This way they can define the weight by which each seed word contributes to the new word's rating[79]. Such systems are based on the assumption that semantic similarity between two words (the likeness of their meaning) can be translated to affective (emotional) similarity .The valence rating was calculated on the continuous scale of $[-1,1]$ at a sentence level (combined through various rules) and the ratings of the individual sentences from which the song was comprised were then averaged to obtain an 'overall lyrics rating' of the sample.

Two different similarity metrics (**table 5.2**) to calculate the seed word's weights as well as various combination rules (**table 5.3**) to calculate the sentence's valence from its individual words were used.

Similarity Metric	Equation	Value Range
Google(-based Semantic) Relatedness	$E(w_i, w_j) = \frac{max\{L\} - \log\lvert D; w_i, w_j\rvert}{\log\lvert D\rvert - min\{L\}}$, **where:** $L = \{\log\lvert D; w_i\rvert, \log\lvert D; w_j\rvert\}$	$[0, \infty]$
(Point-wise) Mutual Information (PMI)	$I(w_i, w_j) = \log \frac{\frac{\lvert D; w_i, w_j\rvert}{\lvert D\rvert}}{\frac{\lvert D; w_i\rvert}{\lvert D\rvert}\frac{\lvert D; w_j\rvert}{\lvert D\rvert}}$	$[-\infty, \infty]$ **I>0 : Similarity I<0 : Dissimilarity I=0 : Independence**

Table 5.2: Similarity metrics to calculate the seed word's similarity

Where: $w_i, \ldots, w_{i+n}$ are query words, $\{D; w_i, \ldots, w_{i+n}\}$ are the set of results $\{D\}$ returned for these query words and the number of documents in each result set is $\lvert D; w_i, \ldots, w_{i+n}\rvert$

Combination Rule	Equation
Min Max	$v(s)=\max_i(\lvert v(w_i)\rvert)\,sign(v(w_z))$, where: $z=\operatorname{argmax}_{(i)}(\lvert v(w_i)\rvert)$
Plain Average	$v(s)=\frac{1}{N}\sum_{i=1}^{N} v(w_i)$
Weighted Average	$v(s)=\frac{1}{N\sum_{i=1}^{N}\lvert v(w_i)\rvert}\sum_{i=1}^{N} v(w_i)^2\,sign(v(w_i))$

Table 5.3: Combination rules to calculate the sentence's valence from its individual words

Where: $s=w_1 w_2 .... w_N$ a sentence and $v(s)$ its valence rating.

All cases were studied for both 200 and 300 seed words.

Finally, we calculated the accuracy of the ratings, by comparing them to the final labels of our samples. The results can be found in **Chapter 7**.

5.5 AM-FM Sound Signal Features

We already described sound as a signal. Another way to view this signal is as a superposition of n amplitude modulated – frequency modulated (AM-FM) signals, that is signals of the form:

$$r_i(t)=a_i(t)\cos\left(\int_0^t f_i(\tau)d\tau\right) \quad \textbf{(Eq 5.5.1)}$$

Where $a_i(t)$ is called instantaneous amplitude and $f_i(t)$ instantaneous frequency.

Consequently, the signal is expressed as: $s(t)=\sum_{i=1}^{n} r_i(t)$ **(Eq 5.5.2)**

There is evidence [69] that such a model is valid for speech signals, and much research has been centered around making use of features derived from it in order to capture their dynamic nature, fine structure and rapid fluctuations [70], [71].

Signals like the one of **Eq 5.5.1** are called 'speech resonances' or 'formants'.They are used for describing the oscillator systems formed by local cavities in the vocal tract, emphasizing certain frequencies and de-emphasizing others during speech production. It stands to reason that it is a promising model to explore in music classification, as well. First of all, songs contain voice and secondly, the fundamental reasons behind the generation of the dynamic phenomena that cause the fine structures and rapid fluctuations in both music and voice are the same: separated and unstable airflow, vortices, etc.

In fact, the musical effects of 'vibrato' and 'tremolo' can be explained by this model. Vibrato is a

musical effect consisting of a regular pulsating change of pitch. Vibrato is typically characterized in terms of two factors: the amount of pitch variation ("extent of vibrato") and speed with which the pitch is varied ("rate of vibrato") [72]. So, in other words it is a variation in frequency. Tremolo, on the other hand, is a variation in amplitude (**fig. 5.6**).

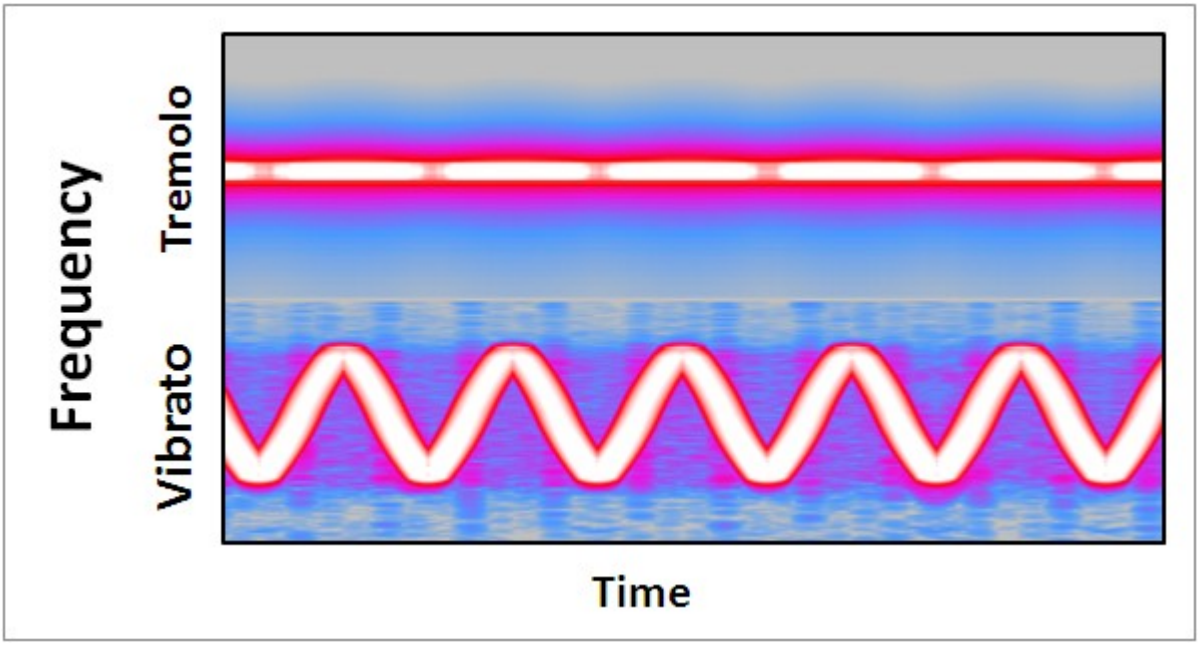


Figure 5.6: Spectrogram illustrating the difference between tremolo and vibrato. Vibrato is a variation in frequency. Tremolo is a variation in amplitude, frequency does not change (Image from: http://en.wikipedia.org/wiki/File:Vibrato_and_tremolo_graph.PNG)

So in our study decided to study this model and derive some features from it to evaluate their effectiveness. We shall describe now the necessary steps in order to obtain them.

5.5.1 Frequency Modulation Percentages (FMPs)

Frequency Modulation Percentages (FMPs) can partially capture the fluctuation of frequencies during a single pitch period.

They are defined as:

$$FMP_i = \frac{B_i}{F_i} \quad \textbf{(Eq 5.5.3)}$$

Where:

$$F_i = \frac{\int_0^T f_i(t)\, a^2{}_i(t)\, dt}{\int_0^T a^2{}_i(t)\, dt} \quad \textbf{(Eq 5.5.4)}$$

and

$$B_i = \frac{\int_0^T [\dot{a}_i^2(t) + (f_i(t) - F_i)^2 a^2{}_i(t)]dt}{\int_0^T a_i^2(t)dt} \quad \textbf{(Eq 5.5.5)}$$

$i = 1,\ldots,n$ is the formant index and T is the time window length.

F_i and B_i are called weighted mean frequency value and mean bandwidth of the formant *i*.

So, before we can calculate. The FMP_i for all *i*, we need to calculate the F_i and B_i for all *i*. And to do so, we need where $a_i(t)$ and $f_i(t)$ for all *i*. If we had a single AM-FM signal $r_i(t)$, we would simply demodulate it. But now, our signal is comprised by *n* superimposed such signals. How do we separate these individual values?

The method we use is called Multiband Demodulation Analysis (MDA). The steps it entails are the following:

1. We filter the signal $s(t)$ using a filter bank. The number, frequency centers and bandwidths of the filters will be examined later, as they were parameters we tweaked. The output will be the separated into the resonance signals (formants), one for each filter of the filterbank.
2. Then we demodulate the signals and obtain the instantaneous amplitude $|(a_i(t))|$ and the instantaneous frequency $f_i(t)$ for each resonance.

A faster approach would be to combine the filtering and the demodulation into one step. This is done through a process called Energy Separation Algorithm which makes use of the Teager Energy Operator[25] [73] (TEO or 'Ψ operator' which is defined as follows for continuous time signals x:

$$\Psi[x] = \dot{x}^2 - x\ddot{x} \quad \textbf{(Eq 5.5.6)}$$

The Energy Separation Algorithm estimates the instantaneous frequency and amplitude of an FM-AM signal $s(t)$ as:

$$f(t) \approx \frac{1}{2\pi}\sqrt{\left(\frac{\Psi[\dot{s}(t)]}{\Psi[s(t)]}\right)} \quad \textbf{(Eq 5.5.7)}$$

and

$$|a(t)| \approx \frac{\Psi[s(t)]}{\sqrt{(\Psi[\dot{s}(t)])}} \quad \textbf{(Eq 5.5.8)}$$

respectively.

Usually the filters used in conjunction with the TEO are Gabor filters [69]. The impulse response $h(t)$ and frequency response $H(f)$ of a Gabor filter are:

$$h(t) = e^{(-\alpha^2 t^2)}\cos(2\pi vt) \quad \textbf{(Eq 5.5.9)}$$

25.The energy operator tracks the energy (per half unit mass) of the source that produces an oscillation signal $x(t) = A\cos(\omega_c t + \theta)$, when applied to it: $\Psi[A\cos(\omega_c t + \theta)] = (A\omega_c)^2$

The sum of Kinetic and Potential Energy of the oscillator is: $\frac{m\dot{x}^2 + kx^2}{2} = \frac{m}{2}A^2\omega^2{}_c$

and

$$H(f)=\frac{\sqrt{(\pi)}}{2\alpha}e^{-\frac{\pi^2(f-v)^2}{\alpha^2}} \textbf{ (Eq 5.5.10)}$$

respectively, where:

v is the central frequency of the filter chosen equal to the formant frequency F
a is the bandwidth parameter (not the actual bandwidth).
The effective rms bandwidth (erb) of the filter is $2\pi\times rms\,bandwidth$ or equivalently: $\alpha/2\pi$.

When the Ψ operator and the Gabor filtering are combined, we get:

$$\Psi[s(t)*h(t)]=[s(t)*\frac{dh(t)}{dt}]^2-s(t)*h(t)[s(t)*\frac{d^2h(t)}{dt^2}] \textbf{ (Eq 5.5.11)}$$

This process (called 'Gabor ESA') is faster than the simple ESA and provides smoother instantaneous frequency times [71].

This method can be applied to discrete time signals, as well. An expression of the Ψ operator for discrete time signals $s(n)$ would be:

$$\Psi[s(n)]=\frac{s^2(n)-s(n-1)s(n+1)}{T^2} \textbf{ (Eq 5.5.12)}$$

where T is the sampling period of the signal $s(n)$.

We use the process described above to calculate the FMPs. Now let us examine a little deeper the matter of the number, the frequency centers and the bandwidths of the filters comprising the filter bank we used. In all cases, what we refer to from now on as 'bandwidth' should not be confused with the Gabor filter's 'effective bandwidth', which was the actual bandwidth of the filters.

We experimented with 3 types of filter banks:

- Filters based on the Mel Scale
- Filters based on the Bark Scale
- Fractional Octave Filters (a category of constant-Q filters), and in particular:
  - 1 Octave Filterbanks
  - 1/3 Octave Filterbanks
  - 1/4 Octave Filterbanks

Let us examine them in further detail:

Filters based on the Mel Scale: The calculation of the filters' center frequencies was done by **Eq.5.1.1**. Their bandwidths were selected accordingly so as for the filters to be half-overlapping.

Filters based on the Bark Scale: The calculation of the filters' center frequencies and bandwidths was based on the Bark Scale (no overlapping).

A constant-Q filter bank consists of n filters whose bandwidth to central frequency ratio is constant:

$$Q=\frac{bw_i}{f_i}, \forall i \quad \textbf{(Eq 5.5.13)}$$

A fractional octave filter bank is a subclass of the constant-Q filter banks. They divide the frequency range into proportional bandwidths that are a fraction of an octave.

In general, for $1/n$ octave analysis, there are *n* band pass filters per octave[26] such that:

$$\frac{f_H}{f_L}=2^{(\frac{1}{n})} \quad \textbf{(Eq 5.5.14)}$$

and

$$f_{cj+1}=f_{cj}2^{\frac{1}{n}} \quad \textbf{(Eq 5.5.15)}$$

where $1/n$ is called the 'fractional bandwidth resolution' and f_L and f_H are the lower and upper cutoff frequencies of a band-pass filter.

The equation below defines the center frequency of each fractional filter:

$$fc(i)=2^{(i/n)} \quad \textbf{(Eq 5.5.16)}$$

The low and high band edge frequencies of each filter can be calculated based on the frequency ratio, and the fractional octave resolution *n*:

$$f_L(i)=f_c(i)2^{\frac{-1}{2n}}, \forall i \quad \textbf{(Eq 5.5.17)}$$

$$f_H(i)=f_c(i)2^{\frac{1}{2n}}, \forall i \quad \textbf{(Eq 5.5.18)}$$

The bandwidth of each filter is: $bw(i)=f_H(i)-f_L(i), \forall i$ **(Eq 5.5.19)**

Our code uses these formulae (**Eq. 5.5.16- Eq. 5.5.19**) as we present them and works for any n, however, we only tested 3 kinds of fractional octave filter banks: 1 octave filterbanks (*n=1*), 1/3 octave filterbanks (*n=3*) and 1/4 octave filterbanks (*n=4*). It would also be interesting to test the results of others, for instance 1/12 octave filterbanks, since each octave consists of 12 chromas.

The corresponding Q values for these 3 filters are shown in **table 5.4**:

Filterbank	Q ratio
1 octave	**1.414**
1/3 octave	**4.318**
1/4 octave	**5.764**

Table 5.4: The fractional octave filterbanks and their corresponding Q ratio.

26.Note that *n* can also be a fraction: $n=1/k$. This, in essence, means that we have 1 filter covering *k* octaves. For instance, an *n* of 1/2 means that we have one filter for every two octaves.

As for the number of bands we used, thus the number of resonance signals of which we considered our signal to be a superposition, it varied as well. The main reason was that the different filterbanks had to cover the same frequency range. However, as n grew larger, more and more bands were required to cover the same frequency range and some memory issues arose.

After we calculated the FMP for each band for each frame (size: 550 samples, half-overlapping) we then proceeded to calculate the following statistic measures of their distribution:

- Their mean
- Their standard deviation
- The mean of the 10% highest values
- The mean of the 10% lowest values

The final size of this feature pack was: $4 \times n$, depending on the number of bands used.

6. CLASSIFICATION PROCESS

We now have a labeled dataset of 412 samples with a number of features extracted from each of its elements. We can move on to training and (subsequently testing the performance of) our classifiers. For this purpose we will use Weka (see **section 3.2**). We decided to classify the samples in one of two classes per dimension.

So, for Valence we have 'Happy' (Positive Valence) and 'Sad' (Negative Valence), and for Activation we have High Activation and Low Activation

6.1 The .arff format

Weka uses an input file of the .arff format (Attribute-Relation File Format). It is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the 'Header' information, which is followed the 'Data' information.

The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. An example header taken from our work is:

```
@RELATION emotion_valence

@ATTRIBUTE num_major_chords_per_duration NUMERIC
@ATTRIBUTE num_minor_chords_per_duration NUMERIC
@ATTRIBUTE num_dominant_7nth_&_suspended_chords_per_duration NUMERIC
@ATTRIBUTE major_chords_duration_ratio NUMERIC
@ATTRIBUTE minor_chords_duration_ratio NUMERIC
@ATTRIBUTE num_dominant_7nth_&_suspended_chords_duration_ratio NUMERIC
@ATTRIBUTE valence {-1,1}
```

The 'Data' section looks like the following example:

```
@DATA
%(Sample: 1) file1 :
0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,1
%(Sample: 2) file10 :
0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,1
%(Sample: 3) file100 :
0.050000,0.200000,0.000000,0.036880,0.148000,0.000000,1
%(Sample: 4) file101 :
0.000000,0.052632,0.000000,0.000000,1.000000,0.000000,-1
%(Sample: 5) file102 :
0.000000,0.200000,0.000000,0.000000,0.790470,0.000000,-1
%(Sample: 6) file103 :
0.000000,0.066667,0.000000,0.000000,0.130490,0.000000,-1
                                    ⋮
%(Sample: 359) file99 :
0.000000,0.600000,0.000000,0.000000,0.730410,0.000000,1
```

Lines that begin with a % are comments. The @RELATION, @ATTRIBUTE and @DATA declarations are case insensitive.

We notice that not all 412 samples were used. This is because samples labeled with a '3' in one dimension, could not be used in this dimension's 2-class classification problem as training examples, since they belonged to neither class.

We ended up using 359 samples for Valence (134 Negative, 225 positive) and 355 samples for Activation (162 Low, 193 High) in most cases.

In **section 7.3** we will see how we can improve our results, by discarding the most 'uncertain' amongst the data, that is, the ones with labels close to the value '3'.

We converted the extracted features, combined with the final labels for each dimension (which we converted to -1 or 1 via a thresholding function[27]) to create 2 separate .arff files (one for Valence and one for Activation) using a matlab script we implemented. The script also checks for inconsistencies in the data due to miscalculations, so in some cases some more samples (up to 10 more) were discarded.

6.2 Feature Selection

In some cases, we used the entire feature vector available for training our models. In other cases, we wanted to select the best features among the ones available, that is the ones that lead to the best possible classification. Weka offers us the tools to perform feature selection on our training set. We made use of the available 'wrapper' techniques based on [74].

When the feature space was small, we used exhaustive search of the feature space (all the combinations of features in it were tested).In that case we obtained the optimal feature set, however, this method's complexity is combinatorial and as the number of features grows, the running time becomes prohibiting.

With larger feature-packs we used the genetic-algorithms-based search method available, described by [75]. It is faster than the exhaustive search, but it does not guarantee finding the optimal subset of features. However, in practice we found that it selected fairly good features.

Finally, in our final results using nearly all features calculated, we made use of the 'Best First' search method for selecting a good feature subset for the Multilayer Perceptron Classifier, as other methods required a lot of running time. This is a greedy method that selects the features that offer the best classification results, starting with an empty set and adding features.(It first selects the best, then the best and another one with which it provides the best results and so on...) As a greedy method it is prone to be attracted to local optimal solutions, and therefore its results are almost certainly suboptimal in practice, however it is still better than not performing any form of feature selection at all and it is fast enough to allow us to improve a little the classification results of the MLP. In all cases we evaluated the classification results of the feature subsets during the feature selection stage using the 10-Fold Cross-Validation method.

k-Fold Cross-Validation in general, can be used instead of dividing our original dataset into a training set and a testing set. It consists of randomly dividing our entire dataset into k equal-sized parts, $X_i, i=1,2,\ldots,k$. We now use one of these parts for validation (testing) and the other $k-1$ for training. We do this k times, each time using another of the parts for validation:

27.Values < 3 were binned to -1, while values >3 were binned to 1.

$$V_1 = X_1 \ , \ T_1 = X_2 \cup X_3 \cup \cdots \cup X_k$$
$$V_2 = X_2 \ , \ T_2 = X_1 \cup X_3 \cup \cdots \cup X_k$$
$$\vdots \qquad\qquad \vdots$$
$$V_k = X_k \ , \ T_2 = X_1 \cup X_2 \cup \cdots \cup X_{k-1}$$

Where T_i is the *i*-th training set and V_i its corresponding validation set.
This method however has two drawbacks [76]. First, in order to allow the training set to be large, we let the validation set be too small. Second, we effectively execute the training step k times, which is more demanding computationally than a simple division of the dataset in training and testing sets. Of course, in our case, the dataset is small, and the running times for $k=10$ were acceptable.

6.3 Model Training & Evaluation

We used the 10-Fold Cross-Validation method for training/evaluating the classifiers in every case.

Training
In **section 2.3** we discussed the theoretical aspects of the classifiers we used during our work. Here we will concentrate on more practical issues. First of all, let us discuss the specific parameters of the classifiers we utilized:

- Naïve Bayes with default parameters
- 3-Nearest Neighbor with default parameters
- Multilayer Perceptron with 2 hidden layers and a learning rate of 0.3

We chose the particular classifiers, because they represented simple[28], well-studied examples easy to implement in a possible application based on our study.

We used this configuration throughout the entire classification stage. Our experimentation with the parameters indicated that generally, 3-NN fared better than 1-NN, 5-NN or 7-NN. Multilayer Perceptron also proved to provide us with better results on average with 2 hidden layers than it did with 1, 3 or the default choice of ' $(attributes + classes)/2$ hidden layers'. However, no particular effort was made to find the optimal configuration for every classification task. We opted for adopting a 'uniform approach' to all tasks, utilizing the same classifiers for all of them, and, although the results were good, further tweaking with their parameters could further improve their performance. As we mentioned in **section 2.3**, 'no single classifier is the best for all problems', so a uniform approach is generally wrong.

Another reason for using these particular parameter values (among the smallest choices possible), is that they lead to smaller training/classification times. Especially in the case of the MLP, this proves to be very important, because the running times for training networks with many hidden layers were very high.

Finally, these three methods represented three fundamentally different approaches to a classification problem. The Naïve Bayes classifier uses a parametric statistical model (though simplified by the assumption of statistical independence among the features). The k-Nearest Neighbor is a non-parametric method, so it is useful if little or no prior knowledge about the distribution of the data is available. Finally, the Multilayer Perceptron uses the entirely different neural network approach.

Voting
This diversity meant that different strengths and weaknesses could be attributed to the classifiers. And we figured that this could be used to our advantage. In the later stages of our work, we decided to

28.'Simpler' does not always imply 'lesser'.

combine all these classifiers together, and use all of them to perform the classification.

We did this through voting[29] . We combined the output of the classifiers d_{ji} into a weighted sum:

$$y_i = \sum_j w_j d_{ji}$$

Where $w_j \geq 0$ and $\sum_j w_j = 1$.

We experimented both with average of probabilities and majority voting as a combination rule to determine the final result. Finally, we tried different combinations like giving more weight to the best classifiers (individually among all others) or excluding the worst (also individually) from the voting. Our results improved even more.

Support Vector Machines with Sequential Minimal Optimization Classifier
During the final classification stages, we also experimented with another classifier: A classifier based on Support Vector Machines (SVMs) using the Sequential Minimal Optimization (SMO) algorithm. This classifier outperformed the other three in most cases.

A support vector machine constructs a hyperplane, which can be used for classification. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. The parameters of the maximum-margin hyperplane are derived by solving an optimization problem. The formulation can result in a large optimization problem, which may be impractical for a large number of classes. The Sequential Minimal Optimization algorithm [77], breaks the problem down into 2-dimensional sub-problems that may be solved analytically, eliminating the need for a numerical optimization algorithm.

They too, represent a different approach to classification and synergize well with the rest of the classifiers in ensembles. In fact, in most cases we obtained our results using an ensemble of all 4 classifiers, with two votes given to the best (individually).

Evaluation
Until now, we have not yet discussed how one evaluates the performance of a classifier. Let us examine the simple 2-class case, as such is the case we study in our work.

Let us assume we have two classes: Class A and Class B. And let Class A be what we call the 'null hypothesis' the 'default' class.

The samples truly belong to one of these two classes. The classifier assigns new samples to one of the two available categories A or B. We can, therefore, see that there are 4 possible results of a sample's classification:

1) The Sample belongs in Class A and was assigned to Class A: In this case, all is fine and we have what we call a 'True Positive' (TP).
2) The Sample belongs in Class B and was assigned to Class A: In this case, we have a misclassification which we shall call a 'Type I error' or 'False Positive' (FP).
3) The Sample belongs in Class B and was assigned to Class B: In this case, all is fine and we have what we call a 'True Negative' (TN).

29.Also known as 'classifier ensemble' or 'linear opinion pool'

4) The Sample belongs in Class A and was assigned to Class B: In this case, we have a misclassification which we shall call a 'Type II error' or a ' False Negative' (FN).

We can now define the following measures for evaluating our entire classification performance:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{(Number\ of\ Correctly\ Classified\ Samples)}{(Total\ Number\ of\ Samples)}$$

And for individual classes *i* (considering each of them as the 'null hypothesis' for the definition of TP, TN, FP, FN) we can also define:

$$Precision_i = [\frac{TP}{TP + FP}]_i = \frac{(Number\ of\ Samples\ Correctly\ Assigned\ \text{to}\ Class\ i)}{(Total\ Number\ of\ Samples\ Assigned\ \text{to}\ Class\ i)}$$

$$Recall_i = [\frac{TP}{TP + FN}]_i = \frac{(Number\ of\ Samples\ Correctly\ Assigned\ \text{to}\ Class\ i)}{(Total\ Number\ of\ Samples\ Belonging\ \text{to}\ Class\ i)}$$

The overall $Precision$ and $Recall$ measures, are the average of the $Precision_i \forall i$ and $Recall_i \forall i$, respectively. In our case, of course $i \in \{A, B\}$.

A Precision score of 1.0 for a class *i* means that every item labeled as belonging to class *i* does indeed belong to class *i* (but says nothing about the number of items from class i that were not labeled correctly) whereas a Recall of 1.0 means that every item from class *i* was labeled as belonging to class *i* (but says nothing about how many other items were incorrectly also labeled as belonging to class *i*).

We can see that Precision and Recall do not have a real value in their own right (however, if seen together they do have).They are usually combined into a single measure, such as the 'F-measure' or 'F-score' [78]. In its general case it is equal to:

$$F_\beta = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \quad \textbf{(Eq 6.3.1)}$$

Usually when we refer to the 'F-measure', we mean the 'F1-measure', a special case of the **Eq 6.3.1** with $\beta = 1$:

$$F_1 = \frac{Precision \times Recall}{Precision + Recall}$$

Which is actually the weighted harmonic mean[30] of Precision and Recall.

The F-measure can be calculated both for a specific class *i* (using $Precision_i$ and $Recall_i$ to calculate it), and as an overall measure of performance by averaging all the individual $F-measure_i$ values, over all *i*.

30.The harmonic mean of the values $x_i, i = 1,2,..,n$ is $H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$, not to be confused with their arithmetic mean: $H = \frac{1}{n} \sum_{i=1}^{n} x_i$.

7. CLASSIFICATION RESULTS

In this section we will present the results of the classification process. They correspond to two distinct 2-class classification tasks (one for each emotional dimension studied).They will be evaluated in terms of the measures we analyzed above. In the first part (**section 7.1**) we will study individual features when such a thing is possible or valid. We will also group them together in feature-packages in cases we find such a grouping to hold some merit. In **section 7.2** we will combine the best features so far and examine their joint results. In **section 7.3**, we will study the behavior of the classifiers as the dataset is limited to ever more 'certain' samples. Throughout this chapter, we will discuss our findings and make some useful observations about the features, the classifiers and the results in general.

The results' tables themselves can be found in the **Appendix** (**tables A.1-A.41**), and references to them will be made throughout this chapter. We mark the best and the worst-performing classifier in every case with blue and orange color respectively. Our main criterion will be the classifier's accuracy, however, F-measure will also be taken into account, as, in some cases, a classifier would classify all the samples to the most prevalent class, achieving a higher accuracy than others but a low F-measure will reflect the extreme case of $Recall_i = 1$, coupled with low $Precision_i$.

So, we should keep in mind that a 'high' accuracy score might not be that high, after all. Unless we specify otherwise, we will have 359 samples for Valence (134 Negative, 225 positive) and 355 samples for Activation (162 Low, 193 High). We can see a strong bias towards positive valence, and a slight one towards high activation. In these cases, making a decision based only on the prior probabilities of each class (i.e.: assigning all samples to the most frequently encountered class), would yield the accuracy percentages shown on **table 7.1**:

	Accuracy of classifying solely based upon the prior probabilities
Valence	**62.6741 %**
Activation	**54.3662 %**

Table 7.1: Accuracy percentage attained in each dimension when classifying all samples to the class with the largest prior probability.

So, in order for a classifier to perform well, it needs to -at least- achieve better classification accuracies than the one shown above for its dimension.

7.1 Individual Features Results

The term 'individual' is used in a loose sense. A particular feature's statistics are all studied in a combined feature pack in most cases. So do features that generally complement one another. In the end, a feature selection will be made to determine the best features among a feature pack.

EEG features: Best Results

These are the classification results of the statistics (Mean, Standard deviation, Mean of the lowest 10% values Mean of the highest 10% values) of the features extracted by the EEG measurements, that is 7 brainwave measurements every 10msec. Only the labels of the individual annotator that performed the annotation while wearing the EEG device were taken into account, so now the number of samples without a label of '3' has decreased in both dimensions. In other words, we have less samples.. We selected the best features among them. The classification did not yield very good results as we can see

in **Tables A.1 - A.2** of the **Appendix**.

Valence: 50.1946% - 68.0934%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **68.4825%**)

Activation: 55.1440% - 65.0206%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **51.8519%**)

Some observations:

- The results were not good, especially in the Valence dimension. A possible reason might be that only one subject participated in the study and only one measurement was taken from them for each song, not enough experiments to cancel out the noise, that is.
- In Activation the results seem a little promising and appear to deem further study.
- The Nearest Neighbor classifiers perform better than the others in this task.

We will not use these features in any further feature-pack.

Lyrics Classification
Here we present the results of the lyrics classification (Valence dimension only). The lyrics ratings (all individual sentences averaged) were compared to the final label assigned to the corresponding sample The detailed results are shown on **Table A.3** of the **Appendix**.

Valence: 30.28% - 41.39%

As we can see the results are not very good. Possible explanations include:

- The annotators were initially instructed to ignore the songs' lyrics and concentrate on the music. Perhaps, they did so to an extent.
- English was not any of the annotators' native tongue, therefore they would be less affected by the songs' lyrics than native speakers, anyway.
- Our decision to take the plain average of the individual sentences' ratings (at least the ones calculated using other rules than the plain average) was not a very correct choice. An evidence for that is that we seem to obtain the best results from the ratings derived by Plain Averaging in the first place.
- The Beatles' songs were (music-wise) rather happy, even if the lyrics were not. Perhaps the music overshadowed the lyrics.

Music-Inspired sound signal features
These are the statistics we calculated from the sound signal features extracted with the use of MIRtoolbox. They are all inspired by music theory and psychoacoustics. They include roughness, fluctuation, key clarity, mode, hcdf and spectral novelty. We present the classification results per dimension of the combined statistics of each of these individual features on **Tables A.4 – A.15** of the **Appendix**.

We can observe in the dimension of **Valence** that:

- Some features, like Roughness, Fluctuation, HCDF and Mode (in order of decreasing performance) perform exceptionally well. **Roughness**, in particular gives us very good results: approximately **76-78% accuracy**.
- Others, like key clarity and spectral novelty do pretty badly.

While in the dimension of Activation:

- Some features, like Roughness, HCDF, and Fluctuation (in order of decreasing performance) perform exceptionally well. **Roughness**, in particular gives us very good results: approximately **72-73% accuracy**.
- Mode, when used as a feature of a Naïve Bayes classifier, is also good but not exceptional.
- Others, like key clarity and spectral novelty do pretty badly.

The results between Valence and Activation are not to be directly compared. Let us not forget that Valence has a significant bias towards one class in our study. So, for example we would be wrong to assume that since roughness leads to a classification of 76-78% accuracy in valence and 72-73% accuracy in Activation it is a better feature for Valence classification than for Activation. In fact, it seems quite the opposite.

However, it is safe to say that in both dimensions features like Roughness, Fluctuation, HCDF and -to a lesser extent- Mode are very successful, while others like Key clarity and Spectral novelty are not that good individually.

Another observation we can make is that in both dimensions Naïve Bayes seems to be the most successful among the three classifiers while 3NN is the least. This happens because most of these features are one-dimensional, so the 'feature independence assumption' of Naïve Bayes, from which stem most of its shortcomings, does not come into play. 3NN on the other hand, must take into account the distance of a new sample to its neighbors. Unfortunately, one dimensional distance, means distance on a line, which does not give much flexibility to 3NN.

Chords features

These are the features we extracted from the chords. They include the number of distinct chords per duration, the most probable key (nominal feature), and 6 other features Number of major chords per duration, Number of minor chords per duration, Number of suspended and dominant seventh chords per duration, Major chords duration ratio, Minor chords duration ratio, Suspended and dominant seventh chords duration ratio) we combined into one feature-pack we called 'Specific Chords Features'. The classification results per dimension of each of these three are shown on **Tables A.16 - A.21** of the **Appendix**.

We can observe that:

- The feature '**Most Probable Key**' performs quite well (**62.9526% - 66.2953%** in **Valence**, **59.4366% - 60.2817%**, in **Activation**). However, since it was a nominal feature (its value is a label, not a number), it would not work well combined with the numeric ones, so we will not use it in further classifications.
- The '**number of distinct chords per duration**', unfortunately did not provide us with very good results (**54.8747% - 60.4457%** in **Valence**, **51.8310% - 57.7465%**, in **Activation**)
- The others combined, appear to be marginally good features on their own, we will later see that when combined with other features from other modalities, they allow us to achieve very good results. This might be because the song excerpts we study contain very few chords, so most of the values of these features are 0. When alone, there tends to be an over-generalization of the 0 values by the classifiers. When combined with other features, they boost their performance by 'adding an extra push' towards a correct classification.
- In the dimension of Activation, we generally obtain better results than in Valence when compared to the corresponding accuracy scores we get if we classify all samples to the class with the highest a-priori probability.

Joint Music-Inspired features: Best Results

These are the results obtained by the classifiers using the best features selected among all the music inspired features (music-inspired sound signal features and specific chords features). We decided to group them together under the label 'Music – Inspired Features', as they are all based on music theory and psychoacoustics. The results are shown both on **Tables A.22 - A.23** of the **Appendix** and on **Tables 7.1.1 – 7.1.2**.

JOINT MUSIC-INSPIRED FEATURES - RESULTS FOR VALENCE
Accuracy of classifying solely based upon the prior probabilities: 62.6741 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	minor_chords_duration_ratio, average_key_clarity, max_summarized_fluctuation, average_hcdf, roughness_std	79.6657 %	0.795	0.797	0.795
Multilayer Perceptron (2 hidden layers)	num_minor_chords_per_duration, max_summarized_fluctuation, average_mode, roughness_std	78.8301 %	0.786	0.788	0.786
3 - Nearest Neighbor	num_minor_chords_per_duration, average_roughness, average_spectral_novelty, max_summarized_fluctuation, mean_summarized_fluctuation, average_hcdf, average_roughness_low	83.5655 %	0.834	0.836	0.833

Table 7.1.1: Results for best features selected among all the music inspired features (music-inspired sound signal features and chords features) in the Valence dimension.

JOINT MUSIC-INSPIRED FEATURES - RESULTS FOR ACTIVATION
Accuracy of classifying solely based upon the prior probabilities: 54.3662 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	minor_chords_duration_ratio, mean_summarized_fluctuation, average_roughness_high	80.8451 %	0.809	0.808	0.807
Multilayer Perceptron (2 hidden layers)	major_chords_duration_ratio, mean_summarized_fluctuation, average_roughness, average_key_clarity, average_roughness_low, average_roughness_high	78.0282 %	0.784	0.780	0.781
3 - Nearest Neighbor	num_major_chords_per_duration, num_minor_chords_per_duration, major_chords_duration_ratio, mean_summarized_fluctuation, average_roughness, roughness_std, average_roughness_low, average_roughness_high	78.5915 %	0.786	0.786	0.786

Table 7.1.2: Results for best features selected among all the music inspired features (music-inspired sound signal features and chords features) in the Activation dimension.

Judging from the results shown on **Table 7.1.1** and **Table 7.1.2**, we can observe that:

- We obtained very good results in both dimensions.
- In Valence, the most prevalent features are: fluctuation (especially its maximum value), roughness and the number and duration of minor chords.
- In Activation, the best features are: roughness, fluctuation (especially its mean value), and the number and duration of minor and major chords.
- We see that some features that were out-shined by others until now, have become useful additions to the prevalent features (key clarity, spectral novelty, specific chords features).
- We see a that the results in Activation (approximately 78-81%) are a very good improvement over the a-priori based classification (54,3%) the results are also very good, but to a lesser extend in Valence (approximately 79-83.5% over the 62.8% of the a-priori based classification)
- MLP classifiers perform slightly worse than the other two, though still well.

MFCCs, MFCC deltas & MFCC accelerations Results

This Feature-pack consists of the various statistics (Mean, Standard deviation, Mean of the lowest 10% values Mean of the highest 10% values) of the MFCCs and their 1st and 2nd order derivatives calculated per frame of each sample. We look at various combinations: only the MFCCs, only the MFCC deltas, only the MFCC accelerations, and all the combinations of the above. The results can be found in detail in **Tables A.24 – A.25**. To summarize them:

Valence: 64.3%-75.5%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **62.6741%**)

Activation: 65.6%-75.2%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **54.3662%**)

The MFCC-related features appear to be generally good features for our classification tasks. By observing **table A.24** and **table A.25** we can notice that:

- The results for Activation seem to be a particularly good improvement over the accuracy we get when classifying solely based upon the prior probabilities (54.4%). In Valence they are also good, but not as impressive.
- In Valence, we get the best results by combining the MFCCs and their 2nd derivatives together. MFCCs alone lead to the worst classification rates.
- In Activation, we get the best results by combining the MFCCs and their 1st derivatives together. MFCC accelerations alone lead to the worst classification rates.
- Although the results are good, the music-inspired features fare noticeably better.

FMPs Individual Statistics

These are the classification results obtained using as features various statistics (Mean, Standard deviation, Mean of the lowest 10% values Mean of the highest 10% values) of the FMPs for the various filterbanks used. We study each feature separately and their respective results are shown on **Tables A.26 - A.35** of the **Appendix**. To summarize them:

Valence: 67.4%-78.2%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **62.6741%**)

Activation: 65%-81.7%
(Accuracy when classifying all samples to the class with the highest a-priori probability: **54.3662%**)

By observing these results, we notice the following:

- The results are very good in both dimensions, sometimes almost as good as the ones obtained by the music-inspired features.
- In most cases the Multilayer Perceptron classifiers work best.
- Usually, the same filterbank and classifier work best for all 4 features.
- Naïve Bayes achieves very good results. This might indicate a fair amount of independence among FMPs of different bands.
- Overall, the Bark filterbank and the 1/3 octave filterbank work best. This was to be expected, as the bark filterbank emulates the ear, while the 1/3 octave filterbank, having its basis in music theory is being used extensively in similar tasks with success.
- The worst performance is obtained by using a Mel filterbank or a ¼ octave filterbank. To be honest this is not fair for the ¼ octave filterbank. As you might notice the small number of bands used (18) fails to encompass the same frequency range as the other 4. They are therefore, at a disadvantage in this comparison. The reason we used 18 bands is that using more caused memory issues.
- The small differences in the number of samples used were due to some 'Inf' values that arose, presumably because of divisions with very small numbers. They did not affect the a-priori probabilities more than 1% in each case.

7.2 Joint Best Features Results

The best collections of features so far were the following three:

- Music-Inspired Features (A combination of sound and chord derived ones): 16 features
- FMPs' Statistics (4 statistic features of the 16 FMPs, one for each band): 64 features
- MFCCs' Stats (4 statistic features of the 13 FMPs and their corresponding 1st and 2nd derivatives): 156 features

Total: 236 features

The numbers describing the features (one might encounter them in **Tables A.38 – A.41**) are their indices in the feature vector. **Table 7.2.1** shows how the numbers correspond to the features:

Index		Feature subset
1	num_major_chords_per_duration	Music-inspired features
2	num_minor_chords_per_duration	
3	num_dominant_7nth_&_suspended_chords_per_duration	
4	major_chords_duration_ratio	
5	minor_chords_duration_ratio	
6	num_dominant_7nth_&_suspended_chords_duration_ratio	
7	average_roughness	
8	average_key_clarity	
9	average_mode	
10	max_summarized_fluctuation	
11	mean_summarized_fluctuation	
12	average_hcdf	
13	roughness_std	
14	average_spectral_novelty	
15	average_roughness_low	
16	average_roughness_high	
17-80	meanFMP[i],stdFMP[i],mean10_prcnt_highFMP[i],mean10_prcnt_lowFMP[i], $i=1,2,\ldots,16$	FMPs Statistics
81-236	meanMFCC[i],stdMFCC[i],mean10_prcnt_highMFCC[i],mean10_prcnt_lowMFCC[i], $i=0,2,\ldots,12$ meanMFCC_de[i],stdMFCC_de[i],mean10_prcnt_highMFCC_de[i], mean10_prcnt_lowMFCC_de[i], $i=0,2,\ldots,12$ meanMFCC_de_de[i],stdMFCC_de_de[i],mean10_prcnt_highMFCC_de_de[i], mean10_prcnt_lowMFCC_de_de[i], $i=0,2,\ldots,12$	MFCCs Statistics

Table 7.2.1: The indices if the features on the feature vector.

These features constituted the final feature pack studied. However, we could not decide over which of the FMP features to include: The ones derived through the Bark filtering, or those derived through the 1/3 Octave filtering? They seemed to be on par with one another. So, we experimented with both. Let

us see what results we obtained:

Entire final feature vector

The classification results for the entire feature vector as we described it above, are shown on **Tables A.34 – A.37** of the **Appendix**.

The first thing we notice is that now we make use of one more classifier: SMO. SMO until now was outperformed by the other three in nearly all cases. However, now SMO seems to yield better results than the others.

We also notice the use of voting. We tried various combinations of voting (All 3 initial classifiers, all 3 + SMO, All 3 + SMO with the best-performing given two votes, All 3 + SMO with the best-performing given two votes and leaving the worst-performing out) and combination rules (Majority voting, Averaging of individual probabilities). The results varied, with the performance generally improving the more classifiers we used.

The results we get by the feature set that included the FMPs obtained through the bark filterbank are slightly better (about 1% better) , and from now on we will use exclusively these. To summarize them, they are:

Valence: 74.5% - 79.3% (for **single classifiers**)
77.4% - 81.1% (using **voting**)
(Accuracy when classifying all samples to the class with the highest a-priori probability: **62.7507%**)

Activation: 75.4% - 80.1% (for **single classifiers**)
79% - 80.4% (using **voting**)
(Accuracy when classifying all samples to the class with the highest a-priori probability: **54.3860%**)

By observing **Tables A.34 – A.37**, we notice the following:

- In most cases the Multilayer Perceptron and SMO classifiers work best.
- The Naïve Bayes classifiers have the worst performance among all the classifiers. This might imply that now the features are not independent and the Naïve Bayes assumption is false.
- Voting improves the results. The best voting scheme appears to be one that includes all the classifiers. Even Naïve Bayes, the weakest individually, appears to add to the value of a classifier ensemble, since it has different strengths and weaknesses than the rest of the classifiers.
- While the results of the classification are good, they are slightly worse than those of the obtained by using only the music-inspired features.

However, there are still tricks we can try. First of all let us select the best features among the ones we have:

Best Features selected from the final feature vector

Now we performed a feature selection on the final feature set. The results we obtained can be seen in further detail on **Tables A.38 – A.41**. We can see the results summarized in **Table 7.2.2** for Valence and **Table 7.2.3** for Activation.

ALL FEATURES: SELECTED FEATURES - RESULTS FOR VALENCE
Accuracy of classifying solely based upon the prior probabilities: 62.7507 %

Classifier	Accuracy	Precision	Recall	F - Measure
Naive Bayes	**79.0831 %**	**0.790**	**0.791**	**0.790**
MLP (2HL)	**83.0946 %**	**0.831**	**0.831**	**0.827**
3-NN	**81.3754 %**	**0.813**	**0.814**	**0.809**
SMO	**83.9542 %**	**0.838**	**0.840**	**0.838**
Classifier Ensembles				
{Naive Bayes , 3-NN, SMO} Average of Probabilities	**84.4575 %**	**0.853**	**0.845**	**0.845**
{Naive Bayes , 3-NN, SMO} Majority Vote	**82.5215 %**	**0.826**	**0.825**	**0.821**
{Naive Bayes , 3-NN, SMOx2} Average of Probabilities	**84.8138 %**	**0.847**	**0.848**	**0.846**
{Naive Bayes , 3-NN, SMOx2} Majority Vote	**83.3138 %**	**0.833**	**0.833**	**0.833**
{Naive Bayes , 3-NN,MLP (2HL), SMOx2} Average of Probabilities	**85.6734 %**	**0.856**	**0.857**	**0.855**
{Naive Bayes , 3-NN,MLP (2HL), SMOx2} Majority Vote	**85.3868 %**	**0.853**	**0.854**	**0.852**

Table 7.2.2: Classification results for classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Valence Dimension.

ALL FEATURES: SELECTED FEATURES - RESULTS FOR ACTIVATION
Accuracy of classifying solely based upon the prior probabilities: 54.3860 %

Classifier	Accuracy	Precision	Recall	F - Measure
Naive Bayes	**82.1637 %**	**0.835**	**0.822**	**0.822**
MLP (2HL)	**82.4561 %**	**0.827**	**0.825**	**0.825**
3-NN	**81.5789 %**	**0.821**	**0.816**	**0.816**
SMO	**85.0877 %**	**0.851**	**0.851**	**0.851**
Classifier Ensembles				
{Naive Bayes , 3-NN, SMO} Average of Probabilities	**84.4575 %**	**0.853**	**0.845**	**0.845**
{Naive Bayes , 3-NN, SMO} Majority Vote	**82.5215 %**	**0.826**	**0.825**	**0.821**
{Naive Bayes , 3-NN, SMOx2} Average of Probabilities	**83.9542 %**	**0.838**	**0.840**	**0.838**
{Naive Bayes , 3-NN, SMOx2} Majority Vote	**84.5272 %**	**0.844**	**0.845**	**0.843**
{Naive Bayes , 3-NN,MLP (2HL), SMOx2} Average of Probabilities	**81.2865 %**	**0.816**	**0.813**	**0.813**
{Naive Bayes , 3-NN,MLP (2HL), SMOx2} Majority Vote	**81.5789 %**	**0.819**	**0.816**	**0.816**

Table 7.2.3: Classification results for classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Activation Dimension.

Some observations:

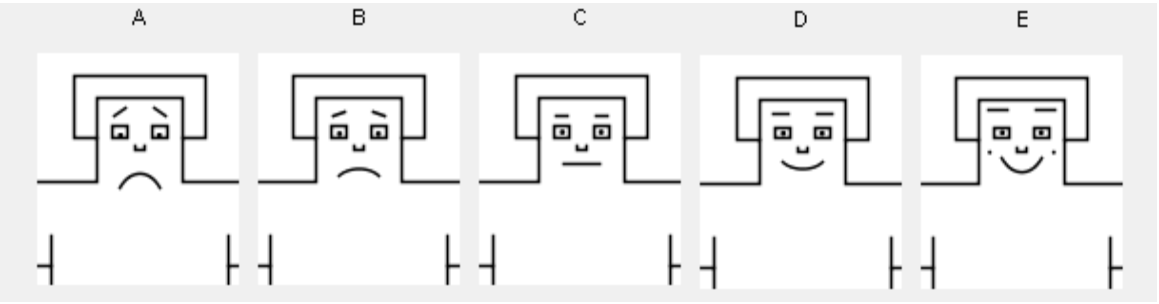
- The feature selection improved our results. They are the best we obtained thus far. The use of the voting techniques has improved even more our classification performance slightly. We have now reached a 85.7% correct classification rate in Valence and 84.5% correct classification rate in Activation.
- Generally, we notice that the more classifiers we add to the ensemble, the better its performance becomes.
- If we observe **Tables** A.34 – A.37, we will see that features from all the feature-packs we combined (music-inspired, MFCCs, FMPs) are selected in the best features' set. Usually the FMP-related features are selected from particular bands (this perhaps deems further study) and so do the MFCC-related ones. Nearly in all cases the zeroth MFCC coefficient's statistics are selected among the best features. This is pretty interesting, since, as we already mentioned $c[0]$ corresponds to the energy of the signal. Finally features from the music-inspired feature-pack are also always selected, especially roughness, fluctuation, duration of minor chords and number of minor chords.

7.3 ROC Analysis

Treatment of the 'uncertain' labels
Until now we used all the labels that had a value other than a '3'.

During the annotation process the possible labels an annotator could assign to a song sample were five in each dimension. Each value corresponds to one of the Self-Assessment Mannequins shown in **fig. 7.3.1**.

Valence Dimension



Activation Dimension

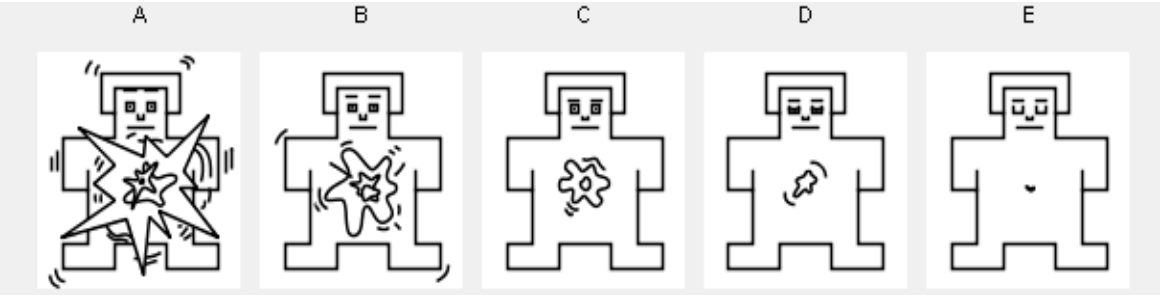


Table 7.3.1: The labels in each of the two dimensions. They were assigned a numeric value (A→1, B→2, C→3, D→4, E→5)

The labels in each of the two dimensions were assigned a numeric value (A→1, B→2, C→3, D→4, E→5) and from this step on they were treated as ordinal data. That is, they were consindered as ordered points on the axis corresponding to their dimension on the Valence-Activation plane.

Three annotators participated in the process. The mean value of the three labels assigned to a song in each dimension were defined as the final labels in that dimension and they were the ones to be actually used in the classification process.

A concentration of the individual annotators' labels is observed towards the center label , i.e. around the median value $\mu_{1/2}=3$. This is evident in both dimensions (**fig 7.3.2 and fig 7.3.3**):

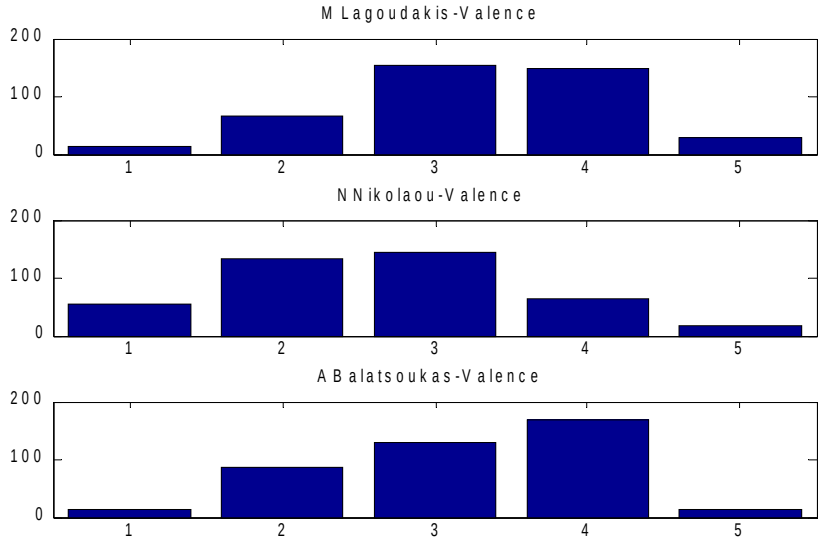


Figure 7.3.2: The distribution of the individual annotators' labels in Valence.

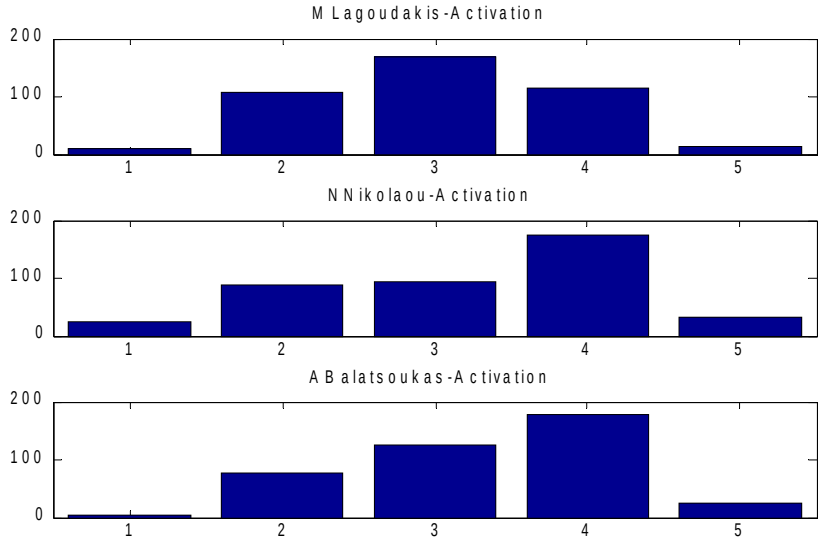


Figure 7.3.3: The distribution of the individual annotators' labels in Activation.

Thus, values near 3 are much more prevalent than the extreme values of $E_L=1$ and $E_U=5$.

One would expect that the final labels taken as the mean value of the corresponding individual labels, will retain the same concentration towards the center of the distribution (again with a median $\mu_{1/2}=3$ and the extreme values $E_L=1$ and $E_U=5$). This is indeed the case, as we can see from their histograms (**fig. 7.3.4 and fig 7.3.5**):

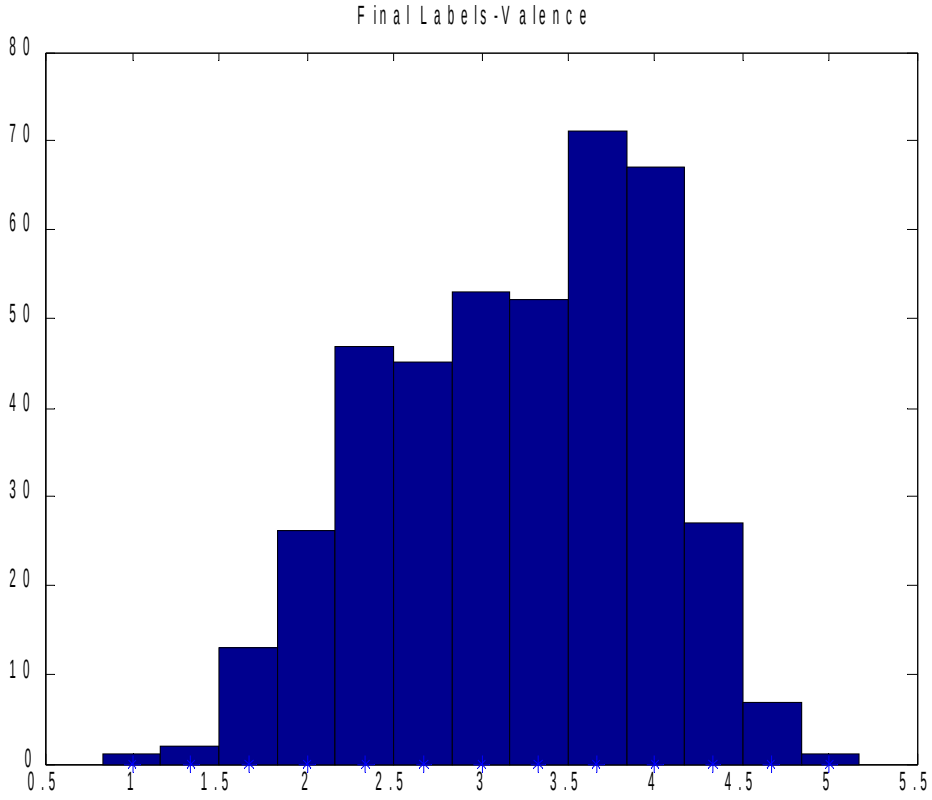


Figure 7.3.4: The distribution of the final labels in Valence.

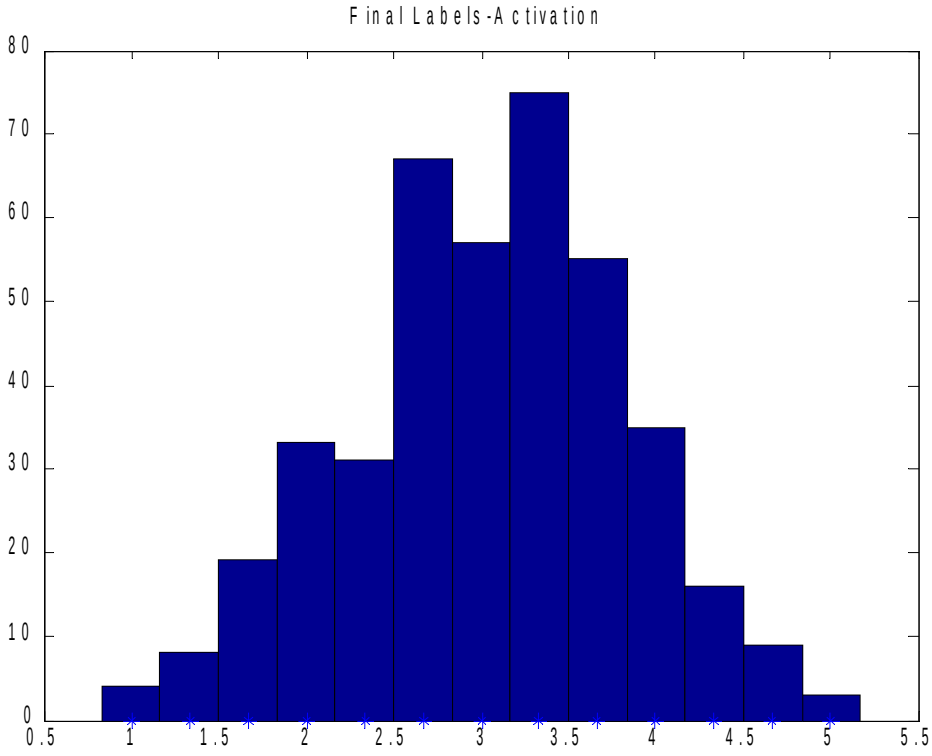


Figure 7.3.5: The distribution of the final labels in Activation.

In order to be able to train and test our classifiers, the target labels have to be meaningful and straightforward. Naturally, the median of the distribution (the value 3) corresponds to song samples for which there is no clear evidence that they are to be considered as 'happy' or 'sad' (in the case of the dimension of valence), or that they are characterized by high or low activation (in the case of the dimension of activation). Therefore, these songs are discarded from our sample as unusable in the classification process.

Taking this thought a step further, it would be a good idea to discard the samples that are relatively "uncertain" prior to the classification step.

We can define an interval centered around the median ($\mu_{1/2}=3$) which corresponds to a "region of uncertainty" R_U .

$$R_U=(B_L, B_U)$$

Where $B_L=\mu_{1/2}-\varepsilon$ the lower bound of the region
and $B_U=\mu_{11/2}+\varepsilon$ the upper bound of the region
with $\varepsilon>0$ being the distance of the bounds from the median

Note: we use the same distance ε in both bounds, in order not to favor one of the two extremes over the other.

A sample with a label belonging in this region R_U is characterized by a low level of certainty as to in which class to be classified, when compared to the samples with labels not falling inside the region. The uncertainty of the songs not belonging in the region decreases as $\varepsilon \to 0$ and increases as ε increases.

The certainty of the songs not belonging in R_U decreases as $\varepsilon \to 0$, since R_U tends towards becoming a single point: $\mu_{1/2}$, on which fall the samples with the maximum uncertainty possible. These are the samples we should discard in any case, as we already mentioned. Still, however the sample contains songs with only a slight tendency toward one of the classes.

Conversely the certainty of the songs not belonging in R_U increases as ε increases, until it becomes absolutely certain where a sample should be classified when $R_U=(E_L, E_U)$, that is all but the songs with labels 1 or 5 (those agreed upon by all annotators as belonging to one of the extremes) are discarded.

Due to the fact that we have calculated the final labels by taking the mean value of all three corresponding individual annotators, the possible values they can take also discrete and they are the following:

$$\{1, 1.3333, 1.6667, 2, 2.3333, 2.6667, 3, 3.3333, 3.6667, 4, 4.3333, 4.6667, 5\}$$

A matlab script was written to iteratively discard the data that fall inside the region of uncertainty. The script was designed to work for any number of annotators but here we examine the case of 3 annotators we ended up with.

Starting by discarding only the samples with final labels that are equal to the median $(\mu_{1/2}=3)$, we iteratively widen the region of uncertainty, discarding the samples with labels whose values fall inside it in every case:

Iteration	Region of Uncertainty (R_U) Bounds	
	Lower Bound (B_L)	Upper Bound (B_U)
1	3	3
2	2.6667	3.3333
3	2.3333	3.6667
4	2	4
5	1.6667	4.3333
6	1.3333	4.6667

Table7.3.1: The upper (B_U) and lower (B_L) bounds of the region of uncertainty (R_U) per iteration of the script described.

By doing so, we obtain different subsets of the dataset (each consecutive subset a subset of the previous one) which contains song samples with an increasing level of certainty as to which class they belong to.

At this point, we must note, that due to the form of the distribution of the final labels (intense concentration of values near the median) the consecutive datasets we obtain from this process contain a dramatically decreasing number of elements. It was indeed very rare that all three annotators agreed that a song deems to be assigned a label of 1 or 5 in any of the two dimensions studied...

We create different .arrf files for each case and visualize the percentage of accuracy of each classifier as a function of the percentage of the initial samples that were discarded.

The results for each dimension studied and for each classifier used are plotted on a Receiver Operating Characteristic Curve (ROC). The ROC curves we obtain (1 figure for each dimensions, with all the classifiers' curves on it) are shown in **figures 7.3.6 – 7.3.9**:

I.ROC Curves for the 'music-inspired features' feature-pack:

Valence Dimension

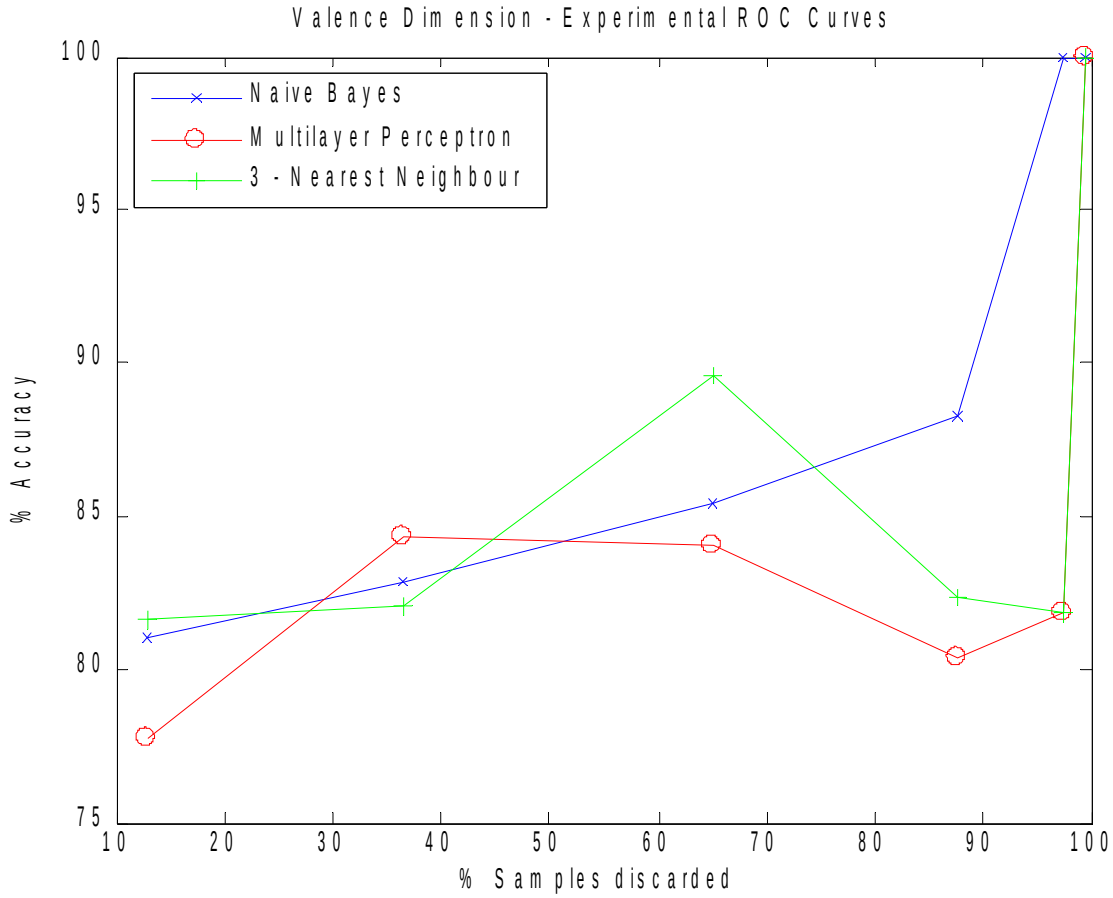


Figure 7.3.6: ROC curve for the 'music-inspired features' in Valence dimension.

Additional Details:

Iteration	Region of Uncertainty (R_U) Bounds		% of initial dataset (412 samples) discarded	% Accuracy			Notes
	Lower Bound ($B_{\_L}$)	Upper Bound (B_U)		Naive Bayes	Multilayer Perceptron (2 Hidden Layers)	3 – Nearest Neighbor	
1	3	3	12.864	81.0585	77.7159	81.6156	
2	2.6667	3.3333	36.4078	82.8244	84.3511	82.0611	
3	2.3333	3.6667	65.0485	85.4167	84.0278	89.5833	
4	2	4	87.6214	88.2353	80.3922	82.3529	
5	1.6667	4.3333	97.3301	100.0000	81.8182	81.8182	
6	1.3333	4.6667	99.5146	100.0000	100.0000	100.0000	*[*]

Table 7.3.2: ROC curve for the 'music-inspired features' in Valence dimension.

*Note: due to the extremely small number of samples (N =3) that remained after discarding all the other values but 1 and 5 (6th iteration), the method of 10-fold cross validation of the dataset could no longer be employed for the evaluation of the classifiers. Instead, we split the dataset into: ⅓ test samples, ⅔ training samples, that is 1 test sample and 2 training samples.

Activation Dimension

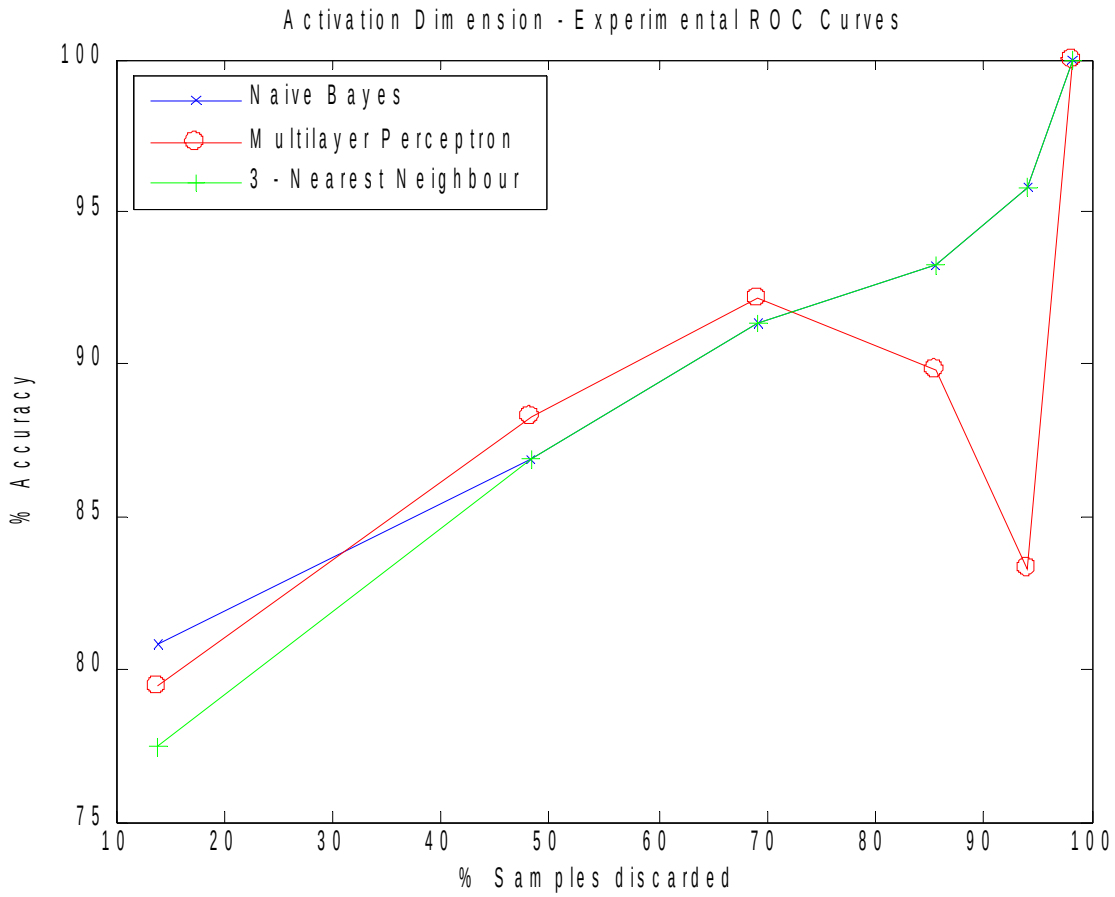


Figure 7.3.7: ROC curve for the 'music-inspired features' in Activation dimension.

Additional Details:

Iteration	Region of Uncertainty (R_U) Bounds		% of initial dataset (412 samples) discarded	% Accuracy			Notes
	Lower Bound (B_L)	Upper Bound (B_U)		Naive Bayes	Multilayer Perceptron (2 Hidden Layers)	3 – Nearest Neighbor	
1	3	3	13.8350	80.8451	79.4366	77.4648	
2	2.6667	3.3333	48.3010	86.8545	88.2629	86.8545	
3	2.3333	3.6667	69.1748	91.3386	92.1260	91.3386	
4	2	4	85.6796	93.2203	89.8305	93.2203	
5	1.6667	4.3333	94.1748	95.8333	83.3333	95.8333	
6	1.3333	4.6667	98.3010	100.0000	100.0000	100.0000	*[*]

Table 7.3.2: ROC curve for the 'music-inspired features' in Activation dimension.

*Note: due to the extremely small number of samples (N = 7) that remained after discarding all the other values but 1 and 5 (6th iteration), the method of 10-fold cross validation of the dataset could no longer be employed for the evaluation of the classifiers. Instead, we split the dataset into: ⅓ test samples, ⅔ training samples, that is: 2 test samples and 5 training samples.

Some Observations:

(1) The Naive Bayes Classifier seems to display the most robust behavior, since by decreasing the uncertainty of the dataset, the accuracy keeps increasing in every case until it reaches 100%, while the Multilayer Perceptron displays the most 'erratic' behavior of the 3 classifiers.

(2) All 3 classifiers eventually reach an accuracy of 100%. However, this is done after all but a handful of extremely distinctive samples of each class are discarded from the dataset, so it is not a very impressive result on its own right.

(3) The classifiers generally work best on subsets of the dataset that are generated by the 2^{nd}, 3^{rd} and 4^{th} iterations of the algorithm that calculates the region of uncertainty. The results are fairly good in these cases ranging from about 80% - 90% accuracy for the dimension of valence and 87% - 93% for the dimension of activation. Moreover there are enough samples in the dataset, to guarantee that these results are fairly consistent.

II.ROC Curves for the selected features from the entire feature-pack:

Valence Dimension

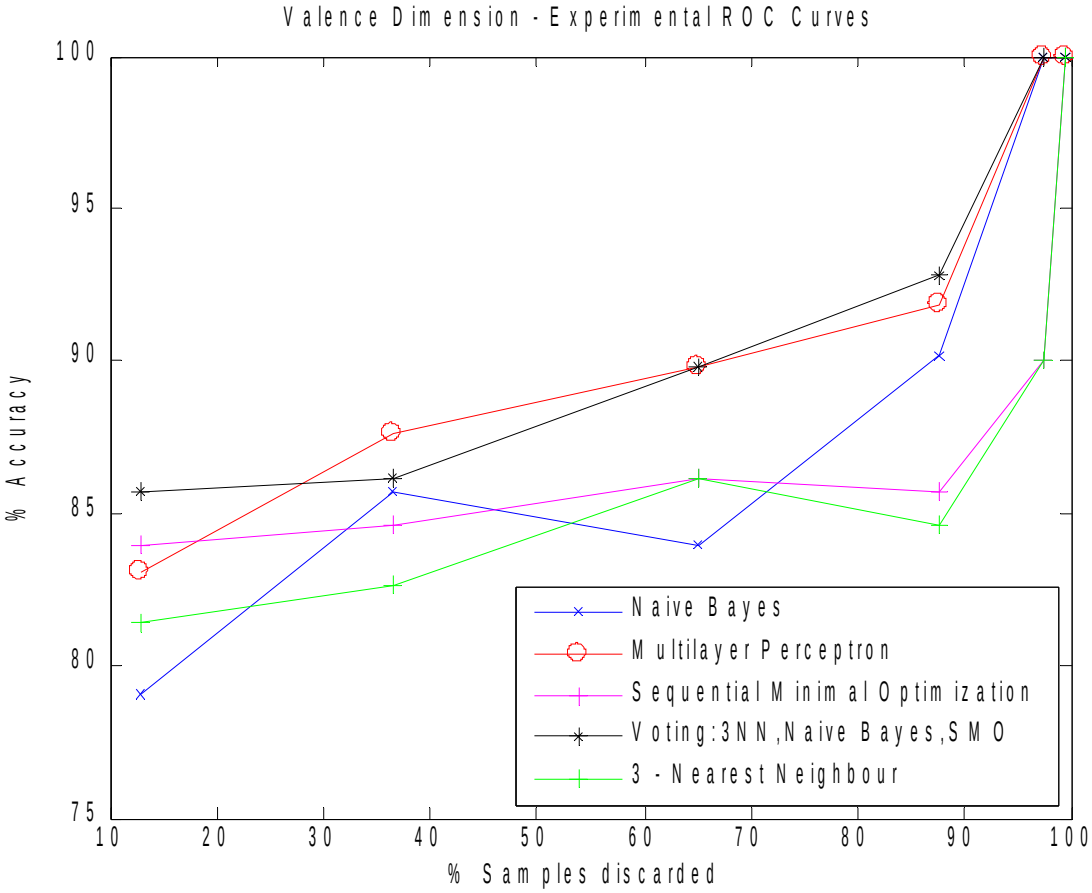


Figure 7.3.8: ROC curve for the best features selected among the entire feature-pack in Valence dimension.

Additional Details:

Iteration	Region of Uncertainty (R_U) Bounds		% of initial dataset (412 samples) discarded	% Accuracy					
	Lower Bound (B_L)	Upper Bound (B_U)		Naive Bayes	MLP	SMO	3NN	Voting{3NN, MLP,NB, SMOx2}	
1	3	3	12.864	79.0831	83.0946	83.9542	81.3754	85.6734	
2	2.6667	3.3333	36.4078	85.7143	87.5912	84.5850	82.6087	86.1660	
3	2.3333	3.6667	65.0485	83.9416	89.7959	86.1314	86.1314	89.7810	
4	2	4	87.6214	90.1478	91.8367	85.7143	84.6327	92.8070	
5	1.6667	4.3333	97.3301	100.0000	100.0000	90.0000	90.0000	100.0000	
6	1.3333	4.6667	99.5146	100.0000	100.0000	100.0000	100.0000	100.0000	*

Table 7.3.3: ROC curve for the best features selected among the entire feature-pack in Valence dimension.

*Note: due to the extremely small number of samples (N = 7) that remained after discarding all the other values but 1 and 5 (6th iteration), the method of 10-fold cross validation of the dataset could no longer be employed for the evaluation of the classifiers. Instead, we split the dataset into: ⅓ test samples, ⅔ training samples, that is: 2 test samples and 5 training samples.

Activation Dimension

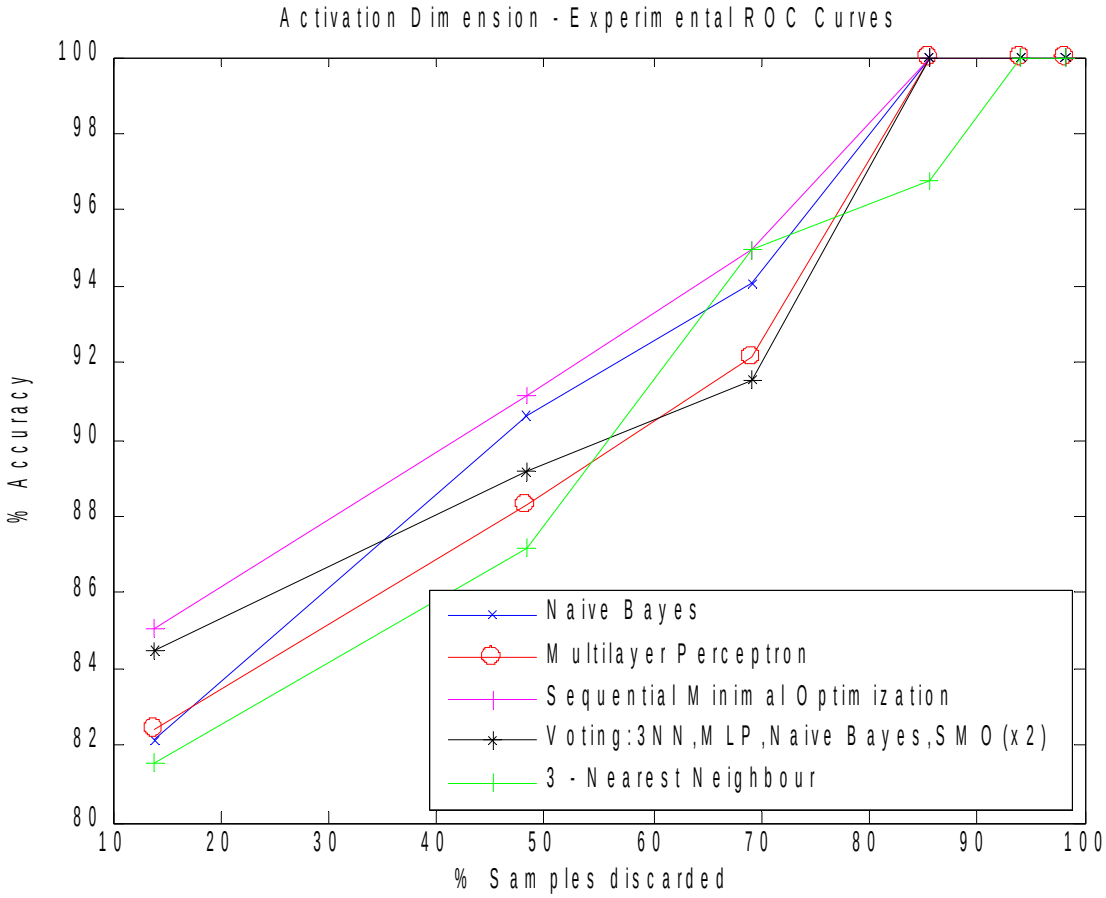


Figure 7.3.9: ROC curve for the best features selected among the entire feature-pack in Activation dimension.

Additional Details:

Iteration	Region of Uncertainty (R_U) Bounds		% of initial dataset (412 samples) discarded	% Accuracy					
	Lower Bound (B_L)	Upper Bound (B_U)		Naive Bayes	MLP	SMO	3NN	Voting{3NN, MLP,NB, SMOx2}	
1	3	3	13.8350	82.1637	82.4561	85.0877	81.5789	84.4575	
2	2.6667	3.3333	48.3010	90.6404	88.2629	91.1330	87.1921	89.1626	
3	2.3333	3.6667	69.1748	94.0678	92.1260	94.9153	94.9153	91.5254	
4	2	4	85.6796	100.0000	100.0000	100.0000	96.7530	100.0000	
5	1.6667	4.3333	94.1748	100.0000	100.0000	100.0000	100.0000	100.0000	
6	1.3333	4.6667	98.3010	100.0000	100.0000	100.0000	100.0000	100.0000	*

Table 7.3.4: ROC curve for the best features selected among the entire feature-pack in Valence dimension.

*Note: due to the extremely small number of samples (N = 7) that remained after discarding all the other values but 1 and 5 (6th iteration), the method of 10-fold cross validation of the dataset could no longer be employed for the evaluation of the classifiers. Instead, we split the dataset into: ⅓ test samples, ⅔ training samples, that is: 2 test samples and 5 training samples.

We see that the models trained with selected features from the entire dataset, are far more robust than the ones trained with only the music-inspired features, as little to no oscillations are observed. By the 3rd iteration we have 83.9-89.8% accuracy for the dimension of valence and 91.5-94.9% for the dimension of activation. And our datasets still consist of 144 and 127 samples respectively, so it is quite an achievement!

Especially in the Activation dimension, we can see that our classifiers perform admirably. By the 4th iteration nearly all of them can classify every sample correctly in a dataset of 60 songs.

The nearest neighbor classifiers seem to be the least successful among our classifiers, however all of them perform very well.

8. CONCLUSIONS & SUGGESTIONS FOR FURTHER WORK

8.1 Conclusions

We examined a number of different features throughout this thesis. Some of them derived from modalities commonly explored when studying music, such as the sound signal itself, while others, were more 'exotic', such as the EEG measurements.

Some of those features turned out to be successful, others not. The EEG - related features did not fare so well. The results in the dimension of Valence ranged from 50.2% to 65.8% correct classification, while the accuracy of classifying solely based upon the prior probabilities was 68.5%. We only had one subject performing the experiment and only once, so this partially explains the bad classification results. On the other hand, some classifiers in the dimension of Activation provided us with slightly positive results (about 65% correct classification rate with a 3NN classifier, while the accuracy of classifying solely based upon the prior probabilities was 52%), giving reason to further study this modality in affective classification tasks.

The lyrics were another field where we did not succeed in obtaining good results. Partly due to the fact that the annotators were given specific instructions to ignore them, partly because they were not native English speakers, partly due to the 'happiness' of The Beatles' music that overshadowed the meaning of the lyrics. But, perhaps, we also paid the consequences of averaging the individual sentences rating to obtain the overall sample's rating, even though the sentences were initially rated using different rules in most cases.

From this point on, we count our victories. The music-inspired features extracted from the sound signal were a good starting point, providing us with up to 78% accuracy for Valence and 73.2% for Activation in some cases (i.e. roughness). The chords features, although not very impressive on their own, when combined with the aforementioned set of features increased our accuracy to up to 83.6% for Valence and 80.8% for Activation.

Next came the MFCCs, which worked best for Activation, with overall good, but not exceptional results on their own. Finally, the FMPs, especially the ones calculated through multiband demodulation using filter banks based on the Bark scale and the 1/3 octave filters provided us with very high classification results.

Finally we decided to combine the best features to obtain a joint feature set of music-inspired features (both from the sound signal and from the chords), FMPs and MFCCs. The overall performance of all classifiers increased slightly, on average.

The next step was to perform a feature selection on this combined feature set and the accuracy of the classifiers using the selected features now exceeded 85%. When combined in ensembles, our classifiers could achieve even 85.6%.

And all this time we only ignored the neutral labels. By selecting ever more certain labels, our classifiers would quickly reach 100%. And although this might be due to the very small size of the dataset in the end, we can achieve 83.9-89.8% accuracy for the dimension of valence and 91.5-94.9% for the dimension of activation with datasets of 144 and 127 samples respectively, so, overall we are content with the results.

We see that between the two sub-problems that we study (classification in the dimension of Valence

and classification in the dimension of Activation), the second one seems to be the 'easiest'. Our classifiers can differentiate quite successfully between songs that belong to high and low Activation. This appears somewhat counter-intuitive. Humans can easily distinguish a happy song from a sad one and they tend to agree more on their classification in the Valence dimension. They consider valence a bipolar dimension, while Activation is somewhat more vague for them. Our classifiers on the other hand seem to do the opposite.

8.2 Suggestions For Further Work

As for what could be done next, even more annotators could lead to a better labeled training set for our classifiers. Special care could also be given to excluding the outliers from the calculation of the final labels.

More features could be explored, such as deriving the chord progressions from the chords files and studying their effect on emotion. Lyrics also deem further study on their own right, and perhaps a model could be devised to fuse together the classification of the music itself and the rating of the lyrics.

As for the classification process, we did not explore all the possibilities regarding the classification techniques and their specific parameters. Perhaps other classifiers could yield even better results.

Another aspect of the procedure that could be improved is the computational complexity of some of the algorithms we use. In a real time application it would be a vital factor. For us it was of secondary importance.

In addition, instead of classifying the overall emotion of a sample, one could try to track the progression of the emotion in time, based upon the analysis of smaller frames of the signal. Such a task would require the use of different techniques, such as Hidden Markov Models, which could capture the dynamic nature of the problem, and associate the emotion in time to that observed during previous times $t-1$, $t-2$, ..., $t-n$, with n as far back as we care (and we can afford computationally) to study.

Finally, one could to try to 'grade' the emotion in each dimension. In other words, instead of classifying to negative and positive valence, one could build classifiers that would classify to one of more than these categories, for example 5 or 10 categories in valence ranging from 'very happy' to 'very sad'.

And once all these techniques offer us a robust and accurate classification in a wider range of categories, perhaps also taking into account the temporal fluctuations of an emotion during a song, and with acceptable computational complexity during the feature extraction and training steps, there could begin to arise real-time applications that make use of them.

9. REFERENCES

1. Qi Lu, Xiaoou Chen, Deshun Yang, Jun Wang : *Boosting For Multi-Modal Music Emotion Classification*, 11th International Society for Music Information Retrieval Conference *(ISMIR 2010)*

2. Laurier, C., Herrera, P.: *Mood Cloud : A Real-Time Music Mood Visualization Tool,* Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain (2008)

3. Juslin, P.N., Laukka, P.: *Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening*, Journal of New Music Research, vol. 33, no. 3 (2004)

4. Benson, D.: *Music: A Mathematical Offering* Cambridge University Press, 3rd Printing (2008)

5. von Hornbostel, Erich M., Curt Sachs: *Zeitschrift für Musik*, 1914

6. von Hornbostel, Erich M., Curt Sachs: *Classification of Musical Instruments: Translated from the Original German by Anthony Baines and Klaus P. Wachsmann, The Galpin Society*, (1961)

7. Bies, David A., and Hansen, C.: *Engineering Noise Control.* (2003)

8. Winckel, F.: *Music, Sound and Sensation: A Modern Exposition*, Dover (1967)

9. Scanavino, C.: *A perceptually grounded approach to sound analysis*, (Tesi di Laurea Magistrale), Politecnico di Torino, III Facoltà di Ingegneria dell'Informazione, Corso di Laurea in Ingegneria Elettronica

10. Olson, Harry F.: *Music, Physics and Engineering,* Dover Publications. p. 249. (1967)

11. Fletcher, H. and Munson, W.A.: *Loudness, its definition, measurement and calculation*. Journal of the Acoustic Society of America 5, 82-108 (1933)

12. William M. Hartmann: *Signals, Sound, and Sensation*, American Institute of Physics (2004)

13. Nave, C. R.: *Loudness Units: Phons and Sones.* HyperPhysics. (2002)

14. Dalebout, S.: *The Praeger Guide to Hearing and Hearing Loss: Assessment, Treatment, and Prevention*, Praeger (2009)

15. Titze IR, Baken RJ, Herzel H: In Titze IR (ed): *Vocal Fold Physiology: Frontiers in Basic Science*, San Diego, CA, Singular Publishing Group, p 143-188, (1993)

16. Curtis Roads:*The Computer Music Tutorial.* The MIT Press, (1996)

17. Zwicker, E.: *Subdivision of the audible frequency range into critical bands*, The Journal of the Acoustical Society of America, 33, Feb., (1961)

18. Gelfand, S.A.: *Hearing- An Introduction to Psychological and Physiological Acoustics* 4th Ed. New York, Marcel Dekker (2004)

19. Walt Kester, MT-*001: Taking the Mystery out of the Infamous Formula, "SNR=6.02N + 1.76dB," and Why You Should Care.* (2009)

20. S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing, vol.1*, Prentice-Hall, (1991)

21. B. Boashash, editor, *Time-Frequency Signal Analysis and Processing – A Comprehensive Reference*, Elsevier Science, Oxford, ISBN 0080443354, (2003)

22. Boretz, Benjamin: *Meta-Variations: Studies in the Foundations of Musical Thought*. Red Hook, New York: Open Space, (1995)

23. Young, R. W.: *Terminology for Logarithmic Frequency Units*. The Journal of the Acoustical Society of America 11 (1): 134–000, (1939)

24. Michael Miller, *The Complete Idiot's Guide to Music, 2nd Ed.*, Penguin Group (USA) Inc. (2005)

25. Tomasz Michal Oliwa: *Genetic algorithms and the abc music notation language for rock music composition*, GECCO '08 Proceedings of the 10th annual conference on Genetic and evolutionary computation (2008)

26. Malandrakis, N., Potamianos, A., Evangelopoulos, G., Zlatintsi, A.,: *A Supervised Approach To Movie Emotion Tracking (ICASSP 2011)*

27. Xiao Hu, J Stephen Downie: *Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata* (2007)

28. Russell, J.A., & Mehrabian, A.: *Evidence for a three-factor theory of emotions*. Journal of Research in Personality, 11, 273-294 (1977)

29. R. Dietz and A. Lang, *Affective agents: Effects of agent affect on arousal, attention, liking and learning*, in Proc. Cognitive Technology Conference (1999)

30. Harry Zhang: *The Optimality of Naive Bayes*, (FLAIRS2004).

31. Rosenblatt, Frank. x.: *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington DC, (1961)

32. Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams.: *Learning Internal Representations by Error Propagation*. David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, (1986)

33. Cybenko, G.: *Approximation by superpositions of a sigmoidal function*. Mathematics of Control, Signals, and Systems, 2(4), 303–314. (1989)

34. Haykin, Simon: *Neural Networks: A Comprehensive Foundation* (2 ed.). Prentice Hall. (1998)

35. Cover TM, Hart PE: *Nearest neighbor pattern classification*, *IEEE Transactions on Information Theory* 13 (1): 21–27. (1967)

36. MATLAB Version 7.10.0.499 (R2010a). The MathWorks Inc. (2010)

37.Olivier Lartillot, Petri Toiviainen, Tuomas Eerola, *A Matlab Toolbox for Music Information Retrieval*, in C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (Eds.), Data Analysis, Machine Learning and

Applications, Studies in Classification, Data Analysis, and Knowledge Organization, Springer-Verlag (2008)

38. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten: *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1 (2009)

39. Boersma, Paul: *Praat, a system for doing phonetics by computer*. Glot International 5:9/10, 341-345. (2001)

40. Florian Eyben, Martin Wöllmer, Björn Schuller: *openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor*, Proc. ACM Multimedia (MM), ACM, Florence, Italy, ISBN 978-1-60558-933-6, pp. 1459-1462, 25.-29.10 (2010)

41. Niedermeyer E. and da Silva F.L.: *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincot Williams & Wilkins, (2004)

42. Tatum, W. O., Husain, A. M., Benbadis, S. R. : *Handbook of EEG Interpretation*, Demos Medical Publishing, (2008)

43. Karl E. Misulis, Toufic Fakhoury: *Spehlmann's Evoked Potential Primer*. Butterworth-heinemann, (2001)

44. P. J. *Lang*.: *Behavioral treatment and bio-behavioral assessment*: *Computer applications,* pp. 119-137, Ablex Publishing, Norwood, NJ, in J.B. Sidowsky et al. (Ed.) Technology in Mental Health Care Delivery Systems (1980)

45. A. Hanjalic, *Extracting moods from pictures and sounds*, *IEEE Signal Processing Magazine*, pp. 90–100 (2006)

46. Krippendorff, K.: *Content Analysis: An Introduction to Its Methodology*, (2nd ed.). Thousand Oaks, CA: Sage, (2004)

47. E. Pampalk, A. Rauber, D. Merkl, *Content-based Organization and Visualization of Music Archives*, ACM Multimedia 2002, pp. 570-579, (2002)

48. Terhardt, E.: *Calculating virtual pitch*, Hearing Research, vol.1, pp. 155-182 (1979)

49. Fletcher, H.: *Auditory Patterns*, The American Physical Society (1949)

50. Fastl H.: *Fluctuation strength and temporal masking patterns of amplitude modulated broadband noise*, Hearing Research, 8, pp. 56-69 (1982)

51. Plomp, R. & Levelt, W.J.M. : *Tonal consonance and critical bandwidth*, Journal of the Acoustical Society of America, Vol. 38, pp. 548–560, (1965)

52. Sethares, W. A.: *Tuning, Timbre, Spectrum, Scale*, Springer-Verlag, (1998)

53. Vassilakis, P. N.: *Perceptual and Physical Properties of Amplitude Fluctuation and their Musical Significance*, Doctoral Dissertation, Los Angeles: University of California, Los Angeles, Systematic Musicology. (2001)

54. Krumhansl, *Cognitive foundations of musical pitch*. Oxford UP, (1990)

55. Gómez, E.: *Tonal description of music audio signal*, Phd Thesis, Universitat Pompeu Fabra, Barcelona (2006)

56. Foote, J. & Cooper, M.: Media Segmentation using Self-Similarity *Decomposition*,. In Proc. SPIE Storage and Retrieval for Multimedia Databases, Vol. 5021, pp. 167-75 (2003)

57. Harte, C. A. and M. B. Sandler: *Detecting harmonic change in musical audio, in Proceedings of Audio and Music Computing for Multimedia Workshop*, Santa Barbara, CA, (2006)

58. Norton, Michael; Karczub, Denis: *Fundamentals of Noise and Vibration Analysis for Engineers*. Cambridge University Press, (2003)

59. S.B. Davis, and P. Mermelstein, *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, in IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), pp. 357–366, (1980)

60. Stevens, Stanley Smith; Volkman, John; & Newman, Edwin: *A scale for the measurement of the psychological magnitude pitch,* Journal of the Acoustical Society of America 8 (3): 185–190, (1937)

61. Douglas O'Shaughnessy: *Speech communication: human and machine*, Addison-Wesley. p.150, (1987)

62. T. Ganchev, N. Fakotakis, and G. Kokkinakis, *Comparative evaluation of various MFCC implementations on the speaker verification task*, in 10th International Conference on Speech and Computer *(SPECOM 2005),* Vol. 1, pp. 191–194, (2005)

63. Meinard Müller: *Information Retrieval for Music and Motion,* Springer. p.65. (2007)

64. Dimitrios Ververidis and Constantine Kotropoulos: *Emotional speech recognition: Resources, features, methods, and applications*, Elsevier Speech Communication, vol. 48, issue 9, pp. 1162-1181, (2006)

65. Zhou, G., Hansen, J. H. L., Kaiser, J. F.: *Nonlinear feature based classification of speech under stress,* IEEE Trans. Speech and Audio Processing 9 (3), 201–216, (2001)

66. Nwe, T. L., Foo, S. W., De Silva, L. C.: *Speech emotion recognition using hidden Markov models*. Speech Communication 41, 603–623, (2003)

67. Min Xu *et al.: HMM-based audio keyword generation*. In Kiyoharu Aizawa, Yuichi Nakamura, Shin'ichi Satoh. *Advances in Multimedia Information Processing* - PCM 2004: 5th Pacific Rim Conference on Multimedia. Springer (2004)

68. Jinjin Ye: *Speech Recognition Using Time Domain Features from Phase Space Reconstructions*, MSc thesis, (2004)

69. P. Maragos, J. F. Kaiser, and T. F. Quatieri: *Energy separation in signal modulations with application to speech analysis*, IEEE Trans. Signal Process., vol. 41, no. 10, pp. 3024–3051, (1993)

70. A. Potamianos and P. Maragos: *Speech formant frequency and bandwidth tracking using multiband energy demodulation*, *J. Acoust. Soc. Amer.*, vol. 99, no. 6, pp. 3795–3806, Jun. (1996)

71. D. V. Dimitriadis, P. Maragos, A. Potamianos: *Robust AM-FM Features for Speech Recognition*, Signal Processing Letters, IEEE In Signal Processing Letters, IEEE, Vol. 12, No. 9., pp. 621-624, (2005)

72. Sundberg, Johan: *Acoustic and psychoacoustic aspects of vocal vibrato*, STL-QPSR, vol. 35 no. 2-3, pp. 045-068, (1994)

73. Teager, H. M., Teager, S. M.: *Evidence for nonlinear sound production mechanisms in the vocal tract*, (NATO Advanced Study Institute, Series D, vol. 15). Boston, MA: Kluwer (1990)

74. Ron Kohavi, George H. John: *Wrappers for feature subset selection*, Artificial Intelligence, 97(1-2):273-324, (1997)

75. David E. Goldberg: *Genetic algorithms in search, optimization and machine learning,* Addison-Wesley, (1989)

76. Ethem Alpaydin: *Introduction to Machine Learning*, 2nd edition, MIT Press, (2010)

77. Platt, John: *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, (1998)

78. Nancy Chinchor, *MUC-4 Evaluation Metrics*, in Proc. of the Fourth Message Understanding Conference, pp. 22–29, (1992)

79. Nikos Malandrakis, Alexandros Potamianos, Elias Iosif, Shrikanth Narayanan, *Kernel models for affective lexicon creation*, in proceedings of Interspeech 2011

80. P. Tsiakoulis, A. Potamianos, D. Dimitriadis: *Spectral moment features augmented by low order cepstral coeffcients for robust ASR*, IEEE Signal Processing Letters, 17(6), (2010)

APPENDIX: Classification Result Tables

EEG FEATURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: 68.4825 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**Theta_std, Low_Alpha_min, High_Alpha_min**	**65.7588 %**	**0.607**	**0.658**	**0.613**
Multilayer Perceptron (2 hidden layers)	**Delta_min, Delta_max, Theta_min, Low_Alpha_min, High_Alpha_min, High_Alpha_mean, High_Alpha_std, High_Beta_mean, Low_Gamma_max**	**50.1946 %**	**0.508**	**0.502**	**0.505**
3 - Nearest Neighbor	**Low_Alpha_max**	**68.0934 %**	**0.596**	**0.681**	**0.568**

Table A.1: Results for best features selected among all EEG features (4 statistics for each of the 7 brainwaves) in the Valence dimension.

EEG FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 51.8519 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**Delta_min, Delta_max, Theta_min, Theta_mean, High_Beta_max**	**57.6132 %**	**0.594**	**0.576**	**0.540**
Multilayer Perceptron (2 hidden layers)	**Theta_min, Low_Alpha_std, High_Alpha_min, High_Alpha_mean, Low_Beta_min, High_Beta_min**	**55.1440 %**	**0.466**	**0.555**	**0.551**
3 - Nearest Neighbor	**Delta_min, Delta_mean, Low_Beta_max, Low_Beta_std, High_Beta_mean, Low_Gamma_mean, Low_Gamma_std**	**65.0206 %**	**0.650**	**0.650**	**0.650**

Table A.2: Results for best features selected among all EEG features (4 statistics for each of the 7 brainwaves) in the Activation dimension.

LYRICS CLASSIFICATION RESULTS (VALENCE ONLY)

Similarity Metric Used	Combined Using Rule	Seed Words Used	Accuracy
Google Relatedness	Min Max	200	0.3778
Google Relatedness	Min Max	300	0.3028
Google Relatedness	Plain Average	200	0.4139
Google Relatedness	Plain Average	300	0.3111
Google Relatedness	Weighted Average	200	0.4083
Google Relatedness	Weighted Average	300	0.3194
Mutual Information	Min Max	200	0.3806
Mutual Information	Min Max	300	0.3500
Mutual Information	Plain Average	200	0.4056
Mutual Information	Plain Average	300	0.4139
Mutual Information	Weighted Average	200	0.3944
Mutual Information	Weighted Average	300	0.3778

Table A.3: Classification results for the lyrics.

MUSIC-INSPIRED SOUND SIGNAL FEATURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: 62.6741 %

- **Roughness statistics** (Average, Standard Deviation, Average of 50% highest, Average of 50% lowest)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	76.3231 %	0.775	0.763	0.766
Multilayer Perceptron (2 hidden layers)	76.0446 %	0.763	0.760	0.761
3 - Nearest Neighbor	77.9944 %	0.777	0.780	0.778

Table A.4: Results for Roughness in the Valence dimension.

- **Fluctuation statistics** (Maximum, Mean Value)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	71.8663 %	0.712	0.719	0.712
Multilayer Perceptron (2 hidden layers)	71.0306 %	0.702	0.710	0.699
3 - Nearest Neighbor	61.2813 %	0.608	0.613	0.610

Table A.5: Results for Fluctuation in the Valence dimension.

- **Key Clarity Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	62.1170 %	0.485	0.621	0.485
Multilayer Perceptron (2 hidden layers)	56.5460 %	0.544	0.565	0.550
3 - Nearest Neighbor	62.6741 %	0.393	0.393	0.483

Table A.6: Results for Key Clarity in the Valence dimension.

- **Mode Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	66.0167 %	0.648	0.660	0.611
Multilayer Perceptron (2 hidden layers)	64.0669 %	0.623	0.641	0.622
3 - Nearest Neighbor	57.6602 %	0.574	0.577	0.575

Table A.7: Results for Mode in the Valence dimension.

- **HCDF Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	69.9164 %	0.693	0.699	0.676
Multilayer Perceptron (2 hidden layers)	68.8022 %	0.677	0.688	0.674
3 - Nearest Neighbor	64.6240 %	0.632	0.646	0.633

Table A.8: Results for HCDF in the Valence dimension.

- **Spectral Novelty Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	61.5599 %	0.547	0.616	0.516
Multilayer Perceptron (2 hidden layers)	62.1170 %	0.588	0.621	0.578
3 - Nearest Neighbor	59.0529 %	0.578	0.591	0.582

Table A.9: Results for Spectral Novelty in the Valence dimension.

MUSIC-INSPIRED SOUND SIGNAL FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 54.3662 %

- **Roughness statistics** (Average, Standard Deviation, Average of 50% highest, Average of 50% lowest)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	73.2394 %	0.741	0.732	0.725
Multilayer Perceptron (2 hidden layers)	71.8310 %	0.720	0.718	0.719
3 - Nearest Neighbor	73.2394 %	0.741	0.732	0.725

Table A.10: Results for Roughness in the Activation dimension.

- **Fluctuation statistics** (Maximum, Mean Value)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	68.4507 %	0.684	0.685	0.682
Multilayer Perceptron (2 hidden layers)	68.7324 %	0.686	0.687	0.686
3 - Nearest Neighbor	64.2254 %	0.641	0.642	0.641

Table A.11: Results for Fluctuation in the Activation dimension.

- **Key Clarity Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	53.8028 %	0.466	0.538	0.395
Multilayer Perceptron (2 hidden layers)	50.9859 %	0.502	0.510	0.502
3 - Nearest Neighbor	49.0141 %	0.486	0.490	0.487

Table A.12: Results for Key Clarity in the Activation dimension.

- **Mode Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	57.4648 %	0.570	0.575	0.552
Multilayer Perceptron (2 hidden layers)	54.3662 %	0.545	0.544	0.544
3 - Nearest Neighbor	54.3662 %	0.544	0.544	0.544

Table A.13: Results for Mode in the Activation dimension.

- **HCDF Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	70.4225 %	0.704	0.704	0.704
Multilayer Perceptron (2 hidden layers)	69.0141 %	0.693	0.690	0.691
3 - Nearest Neighbor	63.3803 %	0.634	0.634	0.634

Table A.14: Results for HCDF in the Activation dimension.

- **Spectral Novelty Statistics** (Average)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	53.8028 %	0.523	0.538	0.498
Multilayer Perceptron (2 hidden layers)	55.4930 %	0.562	0.555	0.555
3 - Nearest Neighbor	50.7042 %	0.505	0.507	0.506

Table A.15: Results for Spectral Novelty in the Activation dimension.

CHORDS FETURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: 62.6741 %

- **Number of Distinct Chords per Duration**

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**60.4457 %**	**0.425**	**0.604**	**0.477**
Multilayer Perceptron (2 hidden layers)	**58.2173 %**	**0.540**	**0.582**	**0.545**
3 - Nearest Neighbor	**54.8747 %**	**0.542**	**0.549**	**0.545**

Table A.16: Results for Number of distinct Chords per Duration in the Valence dimension.

- **Most Probable Key**

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**66.2953 %**	**0.648**	**0.663**	**0.629**
Multilayer Perceptron (2 hidden layers)	**62.9526 %**	**0.610**	**0.630**	**0.612**
3 - Nearest Neighbor	**66.0167 %**	**0.644**	**0.660**	**0.639**

Table A.17: Results for Most Probable Key in the Valence dimension.

- **Specific Chords Features** (Number of major chords per duration, Number of minor chords per duration, Number of suspended and dominant seventh chords per duration, Major chords duration ratio, Minor chords duration ratio, Suspended and dominant seventh chords duration ratio)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**62.6741 %**	**0.599**	**0.627**	**0.591**
Multilayer Perceptron (2 hidden layers)	**63.7883 %**	**0.638**	**0.638**	**0.638**
3 - Nearest Neighbor	**61.8384 %**	**0.607**	**0.618**	**0.611**

Table A.18: Results for Specific Chords Features in the Valence dimension.

CHORDS FETURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 54.3662 %

- **Number of Distinct Chords per Duration**

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**57.7465 %**	**0.580**	**0.577**	**0.578**
Multilayer Perceptron (2 hidden layers)	**56.3380 %**	**0.566**	**0.563**	**0.564**
3 - Nearest Neighbor	**51.8310 %**	**0.524**	**0.518**	**0.519**

Table A.19: Results for Number of distinct Chords per Duration in the Activation dimension.

- **Most Probable Key**

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**60.2817 %**	**0.602**	**0.603**	**0.602**
Multilayer Perceptron (2 hidden layers)	**59.4366 %**	**0.600**	**0.594**	**0.595**
3 - Nearest Neighbor	**60.2817 %**	**0.604**	**0.603**	**0.603**

Table A.20: Results for Most Probable Key in the Activation dimension.

- **Specific Chords Features** (Number of major chords per duration, Number of minor chords per duration, Number of suspended and dominant seventh chords per duration, Major chords duration ratio, Minor chords duration ratio, Suspended and dominant seventh chords duration ratio)

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	**59.1549 %**	**0.611**	**0.592**	**0.588**
Multilayer Perceptron (2 hidden layers)	**59.7183 %**	**0.601**	**0.597**	**0.598**
3 - Nearest Neighbor	**57.1831 %**	**0.580**	**0.572**	**0.572**

Table A.21: Results for Specific Chords Features in the Activation dimension.

JOINT MUSIC-INSPIRED FEATURES - RESULTS FOR VALENCE
Accuracy of classifying solely based upon the prior probabilities: 62.6741 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	minor_chords_duration_ratio, average_key_clarity, max_summarized_fluctuation, average_hcdf, roughness_std	79.6657 %	0.795	0.797	0.795
Multilayer Perceptron (2 hidden layers)	num_minor_chords_per_duration, max_summarized_fluctuation, average_mode, roughness_std	78.8301 %	0.786	0.788	0.786
3 - Nearest Neighbor	num_minor_chords_per_duration, average_roughness, average_spectral_novelty, max_summarized_fluctuation, mean_summarized_fluctuation, average_hcdf, average_roughness_low	83.5655 %	0.834	0.836	0.833

Table A.22: Results for best features selected among all the music inspired features (music-inspired sound signal features and chords features) in the Valence dimension.

JOINT MUSIC-INSPIRED FEATURES - RESULTS FOR ACTIVATION
Accuracy of classifying solely based upon the prior probabilities: 54.3662 %

Classifier	Best Feature Set	Accuracy	Precision	Recall	F-measure
Naïve Bayes	minor_chords_duration_ratio, mean_summarized_fluctuation, average_roughness_high	80.8451 %	0.809	0.808	0.807
Multilayer Perceptron (2 hidden layers)	major_chords_duration_ratio, mean_summarized_fluctuation, average_roughness, average_key_clarity, average_roughness_low, average_roughness_high	78.0282 %	0.784	0.780	0.781
3 - Nearest Neighbor	num_major_chords_per_duration, num_minor_chords_per_duration, major_chords_duration_ratio, mean_summarized_fluctuation, average_roughness, roughness_std, average_roughness_low, average_roughness_high	78.5915 %	0.786	0.786	0.786

Table A.23: Results for best features selected among all the music inspired features (music-inspired sound signal features and chords features) in the Activation dimension.

MFCC FEATURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: 62.6741 %

Classifier	Used as Features statistics of:	Accuracy	Precision	Recall	F - Measure
Naïve Bayes	MFCCs	67.1309 %	0.658	0.671	0.654
Naïve Bayes	MFCC_DELTAS	71.8663 %	0.712	0.719	0.710
Naïve Bayes	MFCC_ACCELERATIONS	72.1448 %	0.715	0.721	0.714
Naïve Bayes	MFCCs & MFCC_DELTAS	71.5877 %	0.709	0.716	0.709
Naïve Bayes	MFCCs & MFCC_ACCELERATIONS	73.2591 %	0.727	0.733	0.725
Naïve Bayes	MFCC_DELTAS & MFCC_ACCELERATIONS	71.3092 %	0.707	0.713	0.707
Naïve Bayes	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	72.1448 %	0.715	0.721	0.714
MLP (2 HL)	MFCCs	64.3454 %	0.638	0.643	0.640
MLP (2 HL)	MFCC_DELTAS	70.4735 %	0.701	0.705	0.702
MLP (2 HL)	MFCC_ACCELERATIONS	67.9666 %	0.674	0.680	0.676
MLP (2 HL)	MFCCs & MFCC_DELTAS	71.8663 %	0.719	0.719	0.719
MLP (2 HL)	MFCCs & MFCC_ACCELERATIONS	73.5376 %	0.735	0.735	0.735
MLP (2 HL)	MFCC_DELTAS & MFCC_ACCELERATIONS	69.3593 %	0.695	0.694	0.694
MLP (2 HL)	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	70.7521 %	0.706	0.708	0.707
3 – Nearest Neighbor	MFCCs	68.2451 %	0.671	0.682	0.664
3 – Nearest Neighbor	MFCC_DELTAS	72.4234 %	0.719	0.724	0.720
3 – Nearest Neighbor	MFCC_ACCELERATIONS	67.9666 %	0.675	0.680	0.677
3 – Nearest Neighbor	MFCCs & MFCC_DELTAS	75.4875 %	0.751	0.755	0.746
3 – Nearest Neighbor	MFCCs & MFCC_ACCELERATIONS	72.7019 %	0.721	0.727	0.718
3 – Nearest Neighbor	MFCC_DELTAS & MFCC_ACCELERATIONS	70.7521 %	0.700	0.708	0.700
3 – Nearest Neighbor	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	74.6518 %	0.742	0.747	0.740

Table A.24: Results for MFCC related features (4 statistics per coefficient: Mean, Standard deviation, Mean of the lowest 10% values Mean of the highest 10% values) in the Valence dimension.

MFCC FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 54.3662 %

Classifier	Used as Features statistics of:	Accuracy	Precision	Recall	F - Measure
Naïve Bayes	MFCCs	65.6338 %	0.701	0.656	0.648
Naïve Bayes	MFCC_DELTAS	71.5493 %	0.743	0.715	0.713
Naïve Bayes	MFCC_ACCELERATIONS	70.4225 %	0.728	0.704	0.702
Naïve Bayes	MFCCs & MFCC_DELTAS	70.7042 %	0.735	0.707	0.704
Naïve Bayes	MFCCs & MFCC_ACCELERATIONS	72.3944 %	0.749	0.724	0.722
Naïve Bayes	MFCC_DELTAS & MFCC_ACCELERATIONS	70.9859 %	0.735	0.710	0.708
Naïve Bayes	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	71.8310 %	0.744	0.718	0.716
MLP (2 HL)	MFCCs	71.2676 %	0.715	0.713	0.713
MLP (2 HL)	MFCC_DELTAS	68.1690 %	0.682	0.682	0.682
MLP (2 HL)	MFCC_ACCELERATIONS	69.8592 %	0.700	0.699	0.699
MLP (2 HL)	MFCCs & MFCC_DELTAS	75.2113 %	0.754	0.752	0.752
MLP (2 HL)	MFCCs & MFCC_ACCELERATIONS	72.3944 %	0.725	0.724	0.724
MLP (2 HL)	MFCC_DELTAS & MFCC_ACCELERATIONS	70.9859 %	0.735	0.710	0.708
MLP (2 HL)	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	72.1127 %	0.721	0.721	0.721
3 – Nearest Neighbor	MFCCs	67.0423 %	0.679	0.670	0.671
3 – Nearest Neighbor	MFCC_DELTAS	67.8873 %	0.692	0.679	0.678
3 – Nearest Neighbor	MFCC_ACCELERATIONS	65.3521 %	0.656	0.654	0.654
3 – Nearest Neighbor	MFCCs & MFCC_DELTAS	68.7324 %	0.698	0.687	0.687
3 – Nearest Neighbor	MFCCs & MFCC_ACCELERATIONS	69.5775 %	0.702	0.696	0.696
3 – Nearest Neighbor	MFCC_DELTAS & MFCC_ACCELERATIONS	63.0986	0.639	0.631	0.631
3 – Nearest Neighbor	MFCCs & MFCC_DELTAS & MFCC_ACCELERATIONS	70.4225 %	0.717	0.704	0.704

Table A.25: Results for MFCC related features(4 statistics per coefficient: Mean, Standard deviation, Mean of the lowest 10% values Mean of the highest 10% values) in the Activation dimension.

FMP FEATURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: approximately 62 %

- **Mean of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	71.6763 %	0.718	0.717	0.717	346
Naive Bayes	Bark	16	77.0774 %	0.767	0.771	0.768	349
Naive Bayes	1 Octave	5	72.6994 %	0.720	0.727	0.719	328
Naive Bayes	1/3 Octave	16	76.4368 %	0.760	0.764	0.760	348
Naive Bayes	¼ Octave	18	67.8161 %	0.666	0.678	0.665	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	74.8555 %	0.743	0.749	0.742	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	78.2235 %	0.779	0.782	0.780	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	74.2331 %	0.739	0.742	0.740	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	77.2989 %	0.771	0.773	0.764	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	68.3908 %	0.678	0.684	0.647	348
3 – Nearest Neighbor	Mel	16	75.7225 %	0.753	0.757	0.749	346
3 – Nearest Neighbor	Bark	16	75.6447 %	0.752	0.756	0.752	349
3 – Nearest Neighbor	1 Octave	5	73.9264 %	0.735	0.739	0.736	328
3 – Nearest Neighbor	1/3 Octave	16	76.7241 %	0.764	0.767	0.765	348
3 – Nearest Neighbor	¼ Octave	18	68.1034 %	0.669	0.681	0.668	348

Table A.26: Results for Mean FMP in the Valence dimension.

- **Standard Deviation of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	69.0751 %	0.682	0.691	0.661	346
Naive Bayes	Bark	16	72.4928 %	0.723	0.725	0.706	349
Naive Bayes	1 Octave	5	67.3780 %	0.666	0.674	0.625	328
Naive Bayes	1/3 Octave	16	74.4253 %	0.739	0.744	0.737	348
Naive Bayes	¼ Octave	18	69.0265 %	0.680	0.690	0.676	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	70.2312 %	0.695	0.702	0.678	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	72.2063 %	0.722	0.722	0.722	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	70.7317 %	0.699	0.707	0.700	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	76.1494 %	0.758	0.761	0.754	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	68.1416 %	0.671	0.681	0.656	348
3 – Nearest Neighbor	Mel	16	69.9422 %	0.692	0.699	0.694	346
3 – Nearest Neighbor	Bark	16	73.9255%	0.735	0.739	0.736	349
3 – Nearest Neighbor	1 Octave	5	71.0366 %	0.702	0.701	0.702	328
3 – Nearest Neighbor	1/3 Octave	16	74.7126 %	0.742	0.747	0.741	348
3 – Nearest Neighbor	¼ Octave	18	75.2212 %	0.748	0.752	0.747	348

Table A.27: Results for Standard Deviation of each FMP in the Valence dimension.

- **Mean of the 10% Highest Values of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	68.4971 %	0.673	0.685	0.671	346
Naive Bayes	Bark	16	77.9370 %	0.779	0.779	0.779	349
Naive Bayes	1 Octave	5	70.4268 %	0.698	0.704	0.697	328
Naive Bayes	1/3 Octave	16	75.8621 %	0.754	0.759	0.753	348
Naive Bayes	¼ Octave	18	69.9115 %	0.694	0.699	0.674	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	69.3642 %	0.685	0.694	0.686	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	76.7908 %	0.766	0.768	0.767	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	74.6951 %	0.742	0.747	0.743	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	76.4368 %	0.764	0.764	0.753	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	72.2714 %	0.716	0.723	0.713	348
3 – Nearest Neighbor	Mel	16	70.8092 %	0.700	0.708	0.701	346
3 – Nearest Neighbor	Bark	16	75.0716 %	0.746	0.751	0.746	349
3 – Nearest Neighbor	1 Octave	5	73.1707 %	0.726	0.732	0.727	328
3 – Nearest Neighbor	1/3 Octave	16	75.8621 %	0.761	0.759	0.760	348
3 – Nearest Neighbor	¼ Octave	18	71.3864 %	0.707	0.714	0.705	348

Table A.28: Results for the Mean of the 10% Highest Values of Each FMP in the Valence dimension.

- **Mean of the 10% Lowest Values of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	68.4049 %	0.686	0.684	0.633	346
Naive Bayes	Bark	16	73.3524 %	0.737	0.734	0.735	349
Naive Bayes	1 Octave	5	67.0732 %	0.672	0.671	0.607	328
Naive Bayes	1/3 Octave	16	76.1494 %	0.773	0.761	0.764	348
Naive Bayes	¼ Octave	18	69.3215 %	0.684	0.693	0.675	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	67.4847 %	0.685	0.675	0.608	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	73.9255 %	0.734	0.739	0.734	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	65.5488 %	0.643	0.655	0.584	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	77.5862 %	0.772	0.776	0.771	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	68.4366 %	0.673	0.684	0.669	348
3 – Nearest Neighbor	Mel	16	65.0307 %	0.637	0.650	0.640	346
3 – Nearest Neighbor	Bark	16	72.4928 %	0.719	0.725	0.720	349
3 – Nearest Neighbor	1 Octave	5	61.8902 %	0.606	0.619	0.610	328
3 – Nearest Neighbor	1/3 Octave	16	75.5747 %	0.751	0.756	0.751	348
3 – Nearest Neighbor	¼ Octave	18	66.9617 %	0.656	0.670	0.647	348

Table A.29: Results for the Mean of the 10% Lowest Values of Each FMP in the Valence dimension.

FMP FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: approximately 54 %

- **Mean FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	75.2212 %	0.773	0.752	0.751	339
Naive Bayes	Bark	16	78.9474 %	0.789	0.789	0.789	342
Naive Bayes	1 Octave	5	70.9375 %	0.747	0.709	0.702	320
Naive Bayes	1/3 Octave	16	80.6452 %	0.817	0.806	0.806	341
Naive Bayes	¼ Octave	18	71.5543 %	0.715	0.716	0.714	341
Multilayer Perceptron (2 Hidden Layers)	Mel	16	81.7109 %	0.818	0.817	0.817	339
Multilayer Perceptron (2 Hidden Layers)	Bark	16	77.4854 %	0.777	0.775	0.775	342
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	75.3125 %	0.753	0.753	0.753	320
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	76.5396 %	0.775	0.765	0.766	341
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	69.5015 %	0.695	0.695	0.692	341
3 – Nearest Neighbor	Mel	16	73.4513 %	0.736	0.735	0.735	339
3 – Nearest Neighbor	Bark	16	80.1170 %	0.802	0.801	0.801	342
3 – Nearest Neighbor	1 Octave	5	75.3125 %	0.753	0.753	0.753	320
3 – Nearest Neighbor	1/3 Octave	16	75.3666 %	0.754	0.754	0.754	341
3 – Nearest Neighbor	¼ Octave	18	65.6891 %	0.661	0.657	0.658	341

Table A.30: Results for the Mean of the 10% Lowest Values of Each FMP in the Activation dimension.

- **Standard Deviation of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	66.0767 %	0.676	0.661	0.659	346
Naive Bayes	Bark	16	74.5614 %	0.758	0.746	0.746	349
Naive Bayes	1 Octave	5	66.5625 %	0.678	0.666	0.664	328
Naive Bayes	1/3 Octave	16	74.1935 %	0.750	0.742	0.742	348
Naive Bayes	¼ Octave	18	63.6637 %	0.653	0.637	0.634	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	69.6165 %	0.710	0.696	0.695	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	74.8538 %	0.752	0.749	0.745	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	65.9375 %	0.659	0.659	0.658	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	76.5396 %	0.766	0.765	0.766	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	70.2703 %	0.708	0.703	0.703	348
3 – Nearest Neighbor	Mel	16	65.7817 %	0.657	0.658	0.657	346
3 – Nearest Neighbor	Bark	16	72.2222 %	0.724	0.722	0.723	349
3 – Nearest Neighbor	1 Octave	5	70.3125 %	0.704	0.703	0.703	328
3 – Nearest Neighbor	1/3 Octave	16	74.1935 %	0.744	0.742	0.742	348
3 – Nearest Neighbor	¼ Octave	18	64.2643 %	0.647	0.643	0.643	348

Table A.31: Results for Standard Deviation of each FMP in the Activation dimension.

- **Mean of the 10% Highest Values of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	71.6814 %	0.716	0.717	0.716	346
Naive Bayes	Bark	16	79.8246 %	0.799	0.798	0.798	349
Naive Bayes	1 Octave	5	65.0000 %	0.716	0.650	0.630	328
Naive Bayes	1/3 Octave	16	77.1261 %	0.774	0.771	0.772	348
Naive Bayes	¼ Octave	18	67.5676 %	0.676	0.676	0.676	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	73.4513 %	0.734	0.735	0.734	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	75.7310 %	0.757	0.757	0.757	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	71.2500 %	0.713	0.713	0.713	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	80.6452 %	0.810	0.806	0.807	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	67.5676 %	0.675	0.676	0.674	348
3 – Nearest Neighbor	Mel	16	73.7463 %	0.738	0.737	0.738	346
3 – Nearest Neighbor	Bark	16	76.0234 %	0.760	0.760	0.760	349
3 – Nearest Neighbor	1 Octave	5	65.9375 %	0.660	0.659	0.660	328
3 – Nearest Neighbor	1/3 Octave	16	75.3666 %	0.754	0.754	0.754	348
3 – Nearest Neighbor	¼ Octave	18	69.6697 %	0.699	0.697	0.697	348

Table A.32: Results for the Mean of the 10% Highest Values of Each FMP in the Activation dimension.

- **Mean of the 10% Lowest Values of Each FMP**

Classifier	Filterbank Used	Number of Bands	Accuracy	Precision	Recall	F - Measure	Samples
Naive Bayes	Mel	16	60.6250 %	0.605	0.606	0.604	346
Naive Bayes	Bark	16	78.0702 %	0.781	0.781	0.780	349
Naive Bayes	1 Octave	5	63.1250 %	0.631	0.631	0.628	328
Naive Bayes	1/3 Octave	16	78.0059 %	0.781	0.780	0.779	348
Naive Bayes	¼ Octave	18	70.2703 %	0.702	0.703	0.702	348
Multilayer Perceptron (2 Hidden Layers)	Mel	16	60.3125 %	0.603	0.603	0.599	346
Multilayer Perceptron (2 Hidden Layers)	Bark	16	75.1462 %	0.755	0.751	0.752	349
Multilayer Perceptron (2 Hidden Layers)	1 Octave	5	59.3750 %	0.609	0.645	0.626	328
Multilayer Perceptron (2 Hidden Layers)	1/3 Octave	16	75.6598 %	0.756	0.757	0.756	348
Multilayer Perceptron (2 Hidden Layers)	¼ Octave	18	68.7688 %	0.687	0.688	0.686	348
3 – Nearest Neighbor	Mel	16	66.5625 %	0.668	0.666	0.666	346
3 – Nearest Neighbor	Bark	16	73.0994 %	0.730	0.731	0.730	349
3 – Nearest Neighbor	1 Octave	5	62.5000 %	0.627	0.625	0.625	328
3 – Nearest Neighbor	1/3 Octave	16	72.4340 %	0.724	0.724	0.722	348
3 – Nearest Neighbor	¼ Octave	18	69.9700 %	0.699	0.700	0.700	348

Table A.33: Results for the Mean of the 10% Lowest Values of Each FMP in the Activation dimension.

ALL FEATURES - RESULTS FOR VALENCE

Accuracy of classifying solely based upon the prior probabilities: 62.7507 %

- **Using Bark filterbank-derived FMPs**

Classifier		Accuracy	Precision	Recall	F - Measure
Naive Bayes		74.4986%	0.740	0.745	0.741
MLP (2HL)		79.3696 %	0.794	0.794	0.794
3-NN		77.3639 %	0.771	0.774	0.767
SMO		78.2235 %	0.779	0.782	0.779
Voting Classifiers	Combination Rule				
{Naive Bayes , 3-NN, MLP (2HL)}	Average of Probabilities	77.3639 %	0.770	0.774	0.769
{Naive Bayes , 3-NN, MLP (2HL)}	Majority Vote	78.7966 %	0.785	0.788	0.783
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Average of Probabilities	79.9427 %	0.797	0.799	0.796
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Majority Vote	80.2292 %	0.800	0.802	0.799
{Naive Bayes , 3-NN, MLP (2HL)x2, SMO}	Average of Probabilities	80.5158 %	0.803	0.805	0.803
{Naive Bayes , 3-NN, MLP (2HL)x2, SMO}	Majority Vote	81.0888 %	0.828	0.881	0.854
{3-NN, MLP (2HL), SMO}	Average of Probabilities	80.2292 %	0.800	0.802	0.800
{3-NN, MLP (2HL), SMO}	Majority Vote	81.6619 %	0.815	0.817	0.814
{3-NN, MLP (2HL)x2, SMO}	Average of Probabilities	78.7966 %	0.786	0.788	0.786
{3-NN, MLP (2HL)x2, SMO}	Majority Vote	81.0888 %	0.809	0.811	0.810

Table A.34: Classification results for combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Valence Dimension.

- **Using 1/3 Octave filterbank-derived FMPs**

Classifier		Accuracy	Precision	Recall	F - Measure
Naive Bayes		73.8506 %	0.733	0.739	0.732
MLP (2HL)		77.0115 %	0.772	0.770	0.771
3-NN		79.0230 %	0.787	0.790	0.786
SMO		78.7356 %	0.786	0.787	0.787

Voting Classifiers	Combination Rule				
{Naive Bayes , 3-NN, MLP (2HL)}	Average of Probabilities	78.1609 %	0.778	0.782	0.778
{Naive Bayes , 3-NN, MLP (2HL)}	Majority Vote	78.7356 %	0.784	0.787	0.784
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Average of Probabilities	79.8851 %	0.797	0.799	0.797
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Majority Vote	79.3103 %	0.791	0.793	0.792
{Naive Bayes , 3-NNx2 , MLP (2HL), SMO}	Average of Probabilities	80.4598 %	0.802	0.805	0.802
{Naive Bayes , 3-NNx2 , MLP (2HL), SMO}	Majority Vote	79.0230 %	0.787	0.790	0.787
{3-NN, MLP (2HL), SMO}	Average of Probabilities	77.8736 %	0.777	0.779	0.778
{3-NN, MLP (2HL), SMO}	Majority Vote	78.4483 %	0.783	0.784	0.783
{3-NNx2 , MLP (2HL), SMO}	Average of Probabilities	79.5977 %	0.793	0.796	0.793
{3-NNx2 , MLP (2HL), SMO}	Majority Vote	80.1724 %	0.799	0.802	0.799

Table A.35: Classification results for combined music-inspired features, FMPs (derived using an 1/3 Octave filterbank) and MFCCs in the Valence Dimension.

ALL FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 54.3860 %

- **Using Bark filterbank-derived FMPs**

Classifier		Accuracy	Precision	Recall	F - Measure
Naive Bayes		75.4386 %	0.788	0.754	0.752
MLP (2HL)		77.7778 %	0.778	0.778	0.778
3-NN		77.4854 %	0.788	0.775	0.775
SMO		80.1170 %	0.801	0.801	0.801
Voting Classifiers	Combination Rule				

{Naive Bayes , 3-NN, MLP (2HL)}	Average of Probabilities	78.9474 %	0.806	0.789	0.789
{Naive Bayes , 3-NN, MLP (2HL)}	Majority Vote	79.2398 %	0.809	0.792	0.792
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Average of Probabilities	80.4094 %	0.806	0.804	0.804
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Majority Vote	78.9474 %	0.794	0.789	0.790
{Naive Bayes , 3-NN, MLP (2HL), SMOx2 }	Average of Probabilities	80.4094 %	0.805	0.804	0.804
{Naive Bayes , 3-NN, MLP (2HL), SMOx2 }	Majority Vote	80.4094 %	0.806	0.804	0.804
{3-NN, MLP (2HL), SMO}	Average of Probabilities	80.1170 %	0.802	0.801	0.801
{3-NN, MLP (2HL), SMO}	Majority Vote	80.1170 %	0.803	0.801	0.801
{3-NN, MLP (2HL), SMOx2 }	Average of Probabilities	80.1170 %	0.801	0.801	0.801
{3-NN, MLP (2HL), SMOx2 }	Majority Vote	79.8246 %	0.799	0.798	0.798

Table A.36: Classification results for combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Activation Dimension.

- **Using 1/3 Octave filterbank-derived FMPs**

Classifier		Accuracy	Precision	Recall	F - Measure
Naive Bayes		75.3666 %	0.783	0.754	0.751
MLP (2HL)		74.4868 %	0.778	0.745	0.745
3-NN		75.0733 %	0.775	0.751	0.751
SMO		77.4194 %	0.775	0.774	0.774
Voting Classifiers	**Combination Rule**				
{Naive Bayes , 3-NN, MLP (2HL)}	Average of Probabilities	76.8328 %	0.806	0.789	0.789
{Naive Bayes , 3-NN, MLP (2HL)}	Majority Vote	77.1261 %	0.781	0.771	0.771
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Average of Probabilities	78.5924 %	0.789	0.786	0.786
{Naive Bayes , 3-NN, MLP (2HL), SMO}	Majority Vote	78.0059 %	0.787	0.780	0.780
{Naive Bayes , 3-NN, MLP (2HL), SMOx2 }	Average of Probabilities	80.4094 %	0.805	0.804	0.804

{Naive Bayes , 3-NN, MLP (2HL), SMOx2 }	**Majority Vote**	**78.0059 %**	**0.781**	**0.780**	**0.780**
{3-NN, MLP (2HL), SMO}	**Average of Probabilities**	**77.1261 %**	**0.772**	**0.771**	**0.771**
{3-NN, MLP (2HL), SMO}	**Majority Vote**	**76.2463 %**	**0.763**	**0.762**	**0.763**
{3-NN, MLP (2HL), SMOx2 }	**Average of Probabilities**	**77.4194 %**	**0.775**	**0.774**	**0.774**
{3-NN, MLP (2HL), SMOx2 }	**Majority Vote**	**77.4194 %**	**0.775**	**0.774**	**0.774**

Table A.37: Classification results for combined music-inspired features, FMPs (derived using an 1/3 Octave filterbank) and MFCCs in the Activation Dimension.

ALL FEATURES: SELECTED FEATURES - RESULTS FOR VALENCE
Accuracy of classifying solely based upon the prior probabilities: 62.7507 %

Single Classifiers:

Classifier	Best Features	Accuracy	Precision	Recall	F - Measure
Naive Bayes	**2,5,6,7,9,11,14,15,16,18,21,22, 26,31,32,33,34,36,38,40,41,42,45, 46,49,50,51,56,59,60,61,63,64,65, 67,68,71,74,78,79,80,81,82,83,84, 86,98,101,102,104,105,109,113,117, 118,128,133,138,144,145,149,151, 153,158,163,166,167,171,176,177, 181,182,183,185,186,188,189,190, 191,192,194,198,209,210,213,217, 229,236**	**79.0831 %**	**0.790**	**0.791**	**0.790**
MLP (2HL)	**THIS FEATURE SET IS NOT NECESSARILY THE OPTIMAL AS IT WAS OBTAINED USING A GREEDY METHOD: 10, 43, 49, 68, 77, 103, 117, 135, 191**	**83.0946 %**	**0.831**	**0.831**	**0.827**
3-NN	**2,4,5,8,9,13,14,20,25,27,37, 41,48,49,51,53,66,72,73,74,76, 77,86,89,91,92,94,98,99,102,104, 105,107,110,111,114,124,130,132, 133,136,142,145,148,150,151,152, 160,164,165,166,167,171,176,183, 185,188,189,190,191,193,194,197, 206,209,210,216,217,231,232,235**	**81.3754 %**	**0.813**	**0.814**	**0.809**

SMO	**2,5,11,13,14,15,21,24,25,27,30,35, 36,39,41,42,45,46,51,53,54,55,56, 60,68,70,72,75,79,80,89,90,93,95, 98,99,103,106,108,109,110,111,112, 114,115,117,121,124,127,130,132, 133,134,136,138,140,142,144,145, 148,151,153,154,155,157,158,160, 167,169,174,180,184,185,186,188, 189,191,194,197,201,203,206,210, 216,217,221,225,228,233,236**	**83.9542 %**	**0.838**	**0.840**	**0.838**

Table A.38: Classification results for single classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Valence Dimension.

Using Voting:

Voting Classifiers	Combination Rule	Features	Accuracy	Precision	Recall	F-measure
{Naive Bayes , 3-NN, SMO}	Average of Probabilities	2,5,6,11,12,16,19,21,23,27,30,32, 33,35,39,40,42,43,45,51,53,55,56, 57,58,61,62,64,65,70,71,72,75,79, 80,81,83,84,88,89,91,93,94,95,98, 100,103,104,105,107,113,114,117, 119,120,122,123,124,125,126,127, 128,129,130,133,134,135,136,140, 142,143,147,148,149,150,152,154, 155,156,157,158,159,161,163,164, 165,166,167,169,174,175,178,182, 184,190,194,195,198,203,205,208, 209,210,211,213,214,215,219,221, 222,224,226,231,232,234,236	84.4575 %	0.853	0.845	0.845
{Naive Bayes , 3-NN, SMO}	Majority Vote	2,5,6,7,9,11,12,14,16,17,20,21,24, 25,26,29,30,32,33,35,36,37,40,44, 46,48,49,50,52,55,58,61,66,70,72, 75,77,78,79,89,93,98,99,102,103, 104,105,107,108,110,111,114,117, 128,130,132,133,134,136,140,142, 145,148,150,151,152,158,160,164, 167,169,171,183,185,188,189,191, 194,197,209,210,216,221,225,235, 236	82.5215 %	0.826	0.825	0.821
{Naive Bayes , 3-NN, SMOx2}	Average of Probabilities	2,5,11,13,14,15,21,24,25,27,30,35, 36,39,41,42,45,46,51,53,54,55,56, 60,68,70,72,75,79,80,89,90,93,95, 98,99,103,106,108,109,110,111, 112,114,115,117,121,124,127,130, 132,133,134,136,138,140,142,144, 145,148,151,153,154,155,157,158, 160,167,169,174,180,184,185,186, 188,189,191,194,197,201,203,206,	84.8138 %	0.847	0.848	0.846

		210,216,217,221,225,228,233,236				
{Naive Bayes , 3-NN, SMOx2}	Majority Vote	2,4,5,13,16,20,30,37,41,45,46,48, 51,53,55,57,58,61,62,63,64,65,71, 72,74,75,79,81,83,84,89,90,91,93, 94,95,98,100,105,107,111,113,114, 117,119,120,122,124,127,128,129, 130,133,134,136,140,142,143,147, 149,152,154,155,157,158,159,160, 163,164,165,166,167,180,182,183, 188,190,194,196,203,204,205,208, 209,210,211,213,214,219,221,222, 223,224,226,231,232,234,236	83.3138 %	0.833	0.833	0.833
{Naive Bayes , 3-NN,MLP (2HL), SMOx2}	Average of Probabilities	SAME AS ABOVE, MOST LIKELY NOT OPTIMAL	85.6734 %	0.856	0.857	0.855
{Naive Bayes , 3-NN,MLP (2HL), SMOx2}	Majority Vote	SAME AS ABOVE, MOST LIKELY NOT OPTIMAL	85.3868 %	0.853	0.854	0.852

Table A.39: Classification results for voting classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Valence Dimension.

ALL FEATURES: SELECTED FEATURES - RESULTS FOR ACTIVATION

Accuracy of classifying solely based upon the prior probabilities: 54.3860 %

Single Classifiers:

Classifier	Best Features	Accuracy	Precision	Recall	F - Measure
Naive Bayes	1,2,5,6,7,8,9,10,11,12,13,14,15,17, 18,21,24,26,29,30,31,33,35,36,37,39, 40,43,44,50,52,53,57,59,60,62,63,65, 66,67,73,75,78,79,80,81,83,84,85,86, 100,103,116,119,120,129,130,136,146, 154,155,161,166,171,181,182,183,185, 188,191,192,195,199,203,216,217,218, 221,229,232	82.1637 %	0.835	0.822	0.822
MLP (2HL)	THIS FEATURE SET IS NOT NECESSARILY THE OPTIMAL AS IT WAS OBTAINED USING A GREEDY METHOD: 6,11,53,77,80,129,142	82.4561 %	0.827	0.825	0.825
3-NN	1,2,3,4,7,8,9,10,11,12,15,16, 17,19,21,22,27,29,30,31,32,33, 34,35,39,42,47,49,53,56,57,60, 61,65,66,68,69,70,71,73,74,76, 78,79,80,81,82,83,84,86,87,88, 91,92,94,95,96,97,98,99,100,101, 103,105,106,107,108,109,111,113, 114,116,117,118,121,124,125,126, 128,129,130,134,135,136,143,146, 147,148,149,150,154,155,159,162, 163,166,167,169,170,172,173,175, 176,178,179,181,186,188,189,190, 193,194,195,199,202,204,205,207,	81.5789 %	0.821	0.816	0.816

	209,211,212,214,215,218,220,221, 223,224,227,230,233,234,235,236				
SMO	5,7,9,12,15,19,21,25,26,27,32, 33,42,44,45,46,47,49,52,56,58, 60,64,65,66,68,69,72,73,74,75, 76,77,78,79,80,88,90,91,93,94, 97,99,102,108,109,111,115,117, 120,122,124,127,130,135,136, 137,138,142,143,149,150,152, 153,155,156,158,159,160,161, 163,165,166,168,172,174,176, 178,182,185,186,188,192,194, 195,198,201,208,211,212,213, 214,215,216,218,220,222,223, 225,229,231,232,234	85.0877 %	0.851	0.851	0.851

Table A.40: Classification results for single classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Activation Dimension.

Using Voting:

Voting Classifiers	Combination Rule	Features	Accuracy	Precision	Recall	F-measure
{Naive Bayes, 3-NN, SMO}	Average of Probabilities	2,5,6,11,12,16,19,21,23,27,30,32,33,35,39, 40,42,43,45,51,53,55,56,57,58,61,62,64, 65,70,71,72,75,79,80,81,83,84,88,89,91, 93,94,95,98,100,103,104,105,107,113, 114,117,119,120,122,123,124,125,126, 127,128,129,130,133,134,135,136,140, 142,143,147,148,149,150,152,154,155, 156,157,158,159,161,163,164,165,166, 167,169,174,175,178,182,184,190,194, 195,198,203,205,208,209,210,211,213, 214,215,219,221,222,224,226,231,232, 234,236	84.4575 %	0.853	0.845	0.845
{Naive Bayes, 3-NN, SMO}	Majority Vote	2,5,6,7,9,11,12,14,16,17,20,21,24,25,26,29, 30,32,33,35,36,37,40,44,46,48,49,50,52, 55,58,61,66,70,72,75,77,78,79,89,93,98, 99,102,103,104,105,107,108,110,111,114,117,1 28,130,132,133,134,136,140,142, 145,148,150,151,152,158,160,164,167, 169,171,183,185,188,189,191,194,197, 209,210,216,221,225,235,236	82.5215 %	0.826	0.825	0.821
{Naive Bayes, 3-NN, SMOx2}	Average of Probabilities	2,5,11,13,14,15,21,24,25,27,30,35,36,39,41,42 ,45,46,51,53,54,55,56,60,68,70,72,75,79,80,8 9,90,93,95,98,99,103,106,108,109,110, 111,112,114,115,117,121,124,127, 130,132,133,134,136,138,140,142,144, 145,148,151,153,154,155,157,158,160, 167,169,174,180,184,185,186,188,189, 191,194,197,201,203,206,210,216,217, 221,225,228,233,236	83.9542 %	0.838	0.840	0.838
{Naive Bayes, 3-NN, SMOx2}	Majority Vote	2,4,5,13,16,20,30,37,41,45,46,48,51,53,55, 57,58,61,62,63,64,65,71,72,74,75,79,81, 83,84,89,90,91,93,94,95,98,100,105,107, 111,113,114,117,119,120,122,124,127, 128,129,130,133,134,136,140,142,143, 147,149,152,154,155,157,158,159,160,	84.5272 %	0.844	0.845	0.843

		163,164,165,166,167,180,182,183,188, 190,194,196,203,204,205,208,209,210, 211,213,214,219,221,222,223,224,226, 231,232,234,236				
{Naive Bayes, 3-NN, MLP (2HL), SMOx2}	**Average of Probabilities**	**SAME AS ABOVE, NOT OPTIMAL**	**81.2865 %**	**0.816**	**0.813**	**0.813**
{Naive Bayes, 3-NN, MLP (2HL), SMOx2}	**Majority Vote**	**SAME AS ABOVE, NOT OPTIMAL**	**81.5789 %**	**0.819**	**0.816**	**0.816**

Table A.41: Classification results for voting classifiers using features selected among the combined music-inspired features, FMPs (derived using a Bark filterbank) and MFCCs in the Activation Dimension.