University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

12-2014

Temporal contextual descriptors and applications to emotion analysis.

Haythem Balti
University of Louisville

Recommended Citation

Balti, Haythem, "Temporal contextual descriptors and applications to emotion analysis." (2014). *Electronic Theses and Dissertations.* Paper 1727.
https://doi.org/10.18297/etd/1727

TEMPORAL CONTEXTUAL DESCRIPTORS AND APPLICATIONS TO EMOTION ANALYSIS

By

Haythem Balti
B.Eng., Network and Telecommunication Engineering, Higher School
Of Communications, 2009

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

December 2014

TEMPORAL CONTEXTUAL DESCRIPTORS AND APPLICATIONS TO EMOTION ANALYSIS

By

Haythem Balti
B.Eng., Network and Telecommunication Engineering, Higher School Of Communications, 2009

A Dissertation Approved On

12/02/2014

by the following Dissertation Committee:

Dissertation Director: Dr. Adel Elmaghraby

Dr. Adrian Lauf

Dr. Ahmed Desoky

Dr. Ayman El-Baz

Dr. Eric Rouchka

ABSTRACT

TEMPORAL CONTEXTUAL DESCRIPTORS AND APPLICATIONS TO EMOTION ANALYSIS

Haythem Balti

December 2, 2014

The current trends in technology suggest that the next generation of services and devices allows smarter customization and automatic context recognition. Computers learn the behavior of the users and can offer them customized services depending on the context, location, and preferences.

One of the most important challenges in human-machine interaction is the proper understanding of human emotions by machines and automated systems. In the recent years, the progress made in machine learning and pattern recognition led to the development of algorithms that are able to learn the detection and identification of human emotions from experience. These algorithms use different modalities such as image, speech, and physiological signals to analyze and learn human emotions. In many settings, the vocal information might be more available than other modalities due to widespread of voice sensors in phones, cars, and computer systems in general. In emotion analysis from speech, an audio utterance is represented by an ordered (in time) sequence of features or a multivariate time series. Typically, the sequence is further mapped into a global descriptor representative of the entire utterance/sequence. This descriptor is used for classification and analysis. In classic approaches, statistics are computed over the entire sequence and used as a global descriptor. This often results in the loss of temporal ordering from the original sequence. Emotion is a succession of acoustic events. By discarding the temporal ordering of these events in the mapping, the classic approaches cannot detect acoustic patterns that lead to a certain emotion.

In this dissertation, we propose a novel feature mapping framework. The proposed framework maps temporally ordered sequence of acoustic features into data-driven global descriptors that integrate the temporal information from the original sequence. The framework contains three mapping algorithms. These algorithms integrate the temporal information implicitly and explicitly in the descriptor's representation.

In the first algorithm, the Temporal Averaging Algorithm, we average the data temporally using leaky integrators to produce a global descriptor that implicitly integrates the temporal information from the original sequence.

In order to integrate the discrimination between classes in the mapping, we propose the Temporal Response Averaging Algorithm which combines the temporal averaging step of the previous algorithm and unsupervised learning to produce data driven temporal contextual descriptors.

In the third algorithm, we use the topology preserving property of the Self-Organizing Maps and the continuous nature of speech to map a temporal sequence into an ordered trajectory representing the behavior over time of the input utterance on a 2-D map of emotions. The temporal information is integrated explicitly in the descriptor which makes it easier to monitor emotions in long speeches.

The proposed mapping framework maps speech data of different length to the same equivalent representation which alleviates the problem of dealing with variable length temporal sequences. This is advantageous in real time setting where the size of the analysis window can be variable.

Using the proposed feature mapping framework, we build a novel data-driven speech emotion detection and recognition system that indexes speech databases to facilitate the classification and retrieval of emotions.

We test the proposed system using two datasets. The first corpus is acted. We showed that the proposed mapping framework outperforms the classic approaches while providing descriptors that are suitable for the analysis and visualization of humans emotions in speech data.

The second corpus is an authentic dataset. In this dissertation, we evaluate the performances of our system using a collection of debates. For that purpose, we propose a novel debate collection that is one of the first initiatives in the literature. We show that the proposed system is able to learn human emotions from debates.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Recently, human-machine interaction has received increasing attention from various fields such as artificial intelligence, machine learning, and information retrieval. One of the most important challenges in human-machine interaction is the proper understanding of human emotions by machines and automated systems. The recognition of the internal emotional state of the user by a machine permits a deeper interaction between both parties.

Emotions can be expressed in different modalities such as voice (the tone of a happy person is different from the tone of a sad person), images (facial expressions), video (actions), physiological signals (EEG, EKG signals). The progress made in machine learning and pattern recognition led to the development of algorithms that are able to learn the detection and identification of human emotions from experience [1, 2, 3, 4, 5, 6]. Image is the most used modality in emotion analysis [1]. The extensive research in image processing and facial emotion analysis led to the development of highly accurate systems for the detection and recognition of facial emotions [7, 8, 9]. These systems were tested on standard datasets with many subjects (individuals) and classes (emotional states) [1].

Emotion analysis can be useful in numerous application such as customer service to detect, frustrated users (which can be automatically forwarded to managers), confused/confident speakers in debates, and truth/lies. In these settings, the vocal information might be more accessible than images. Therefore, the need to develop methods to detect and recognize emotion from speech becomes essential for such applications. In contrast to facial emotion, the analysis of emotional speech has not met a wide success due to few challenges [5]. The challenges faced by researchers to address speech emotion analysis can be categorized into three main challenges.

The first challenge is the choice of suitable acoustic features to capture emotion in speech. Various features have been used in speech emotion analysis [1, 3, 10, 4]. These features are typically categorized into three groups namely: Excitation Source, Vocal Tract, and Prosodic Features. The main challenge is choosing the right set of features to represent the different emotional states. Emotion is a complex notion that depends on several factors such as pitch, speech rate, and tone and there is no acoustic feature that can integrate all these factors. Consequently, researchers explored the selection and combination of various acoustic features to improve the representation of human emotions in speech [1, 3, 4, 10].

The second challenge is the design of an appropriate classifier/Learning Model that is able to classify unforseen instances. As discussed above, various acoustic feature extraction algorithms have been proposed in the literature. Those features have different contexts, meanings, and configurations. The main challenge in building a classifier/Learning Model scheme is handling and combining the heterogeneous structure of those acoustic features efficiently.

The third issue is the proper preparation of a corpus of emotional speech for the evaluation of the system's performances. Emotion has a range of meanings, and it is important for the dataset to be clear where it stands in relation to the various emotional states and qualities to which the term is applied. For instance, a person who develops dataset A might have a different point of view about emotions than a person who develops dataset B. Various datasets have been proposed in the literature such as LDC Emotional Prosody Speech and Transcript(English) [2], Berlin

Emotional Database (German) [11], and Danish Emotional Database (Danish) [12]. One of the main challenges in building a speech emotion detection and/or recognition system is generalizing its performance for datasets with different configurations and languages.

With those challenges in mind, we seek to address and propose novel solutions to some of these issues. Although the work in this thesis is mainly focused on emotion analysis from speech, some of the contributions are easily generalized to other type of time series data as long as the same assumptions are met.

The main contribution of this thesis consists of the development of a novel feature mapping framework that maps temporal speech data into global descriptors that integrate the temporal information from the original sequence. In the classic approaches, temporal speech data is mapped into a static global descriptor before analysis and classification. This often results in the loss of temporal ordering from the original sequence. Emotion is a succession of acoustic events. By discarding the temporal ordering of these events in the mapping, the classic approaches cannot detect acoustic patterns that lead to a certain emotion.

The proposed framework includes three mapping algorithms. These algorithms integrate the temporal information implicitly and explicitly in the descriptor's representation.

In the first algorithm, the Temporal Averaging Algorithm, the data is averaged using leaky integrators [13] to produce a global descriptor that implicitly integrates the temporal information of the original sequence.

In order to integrate the discrimination between classes in the mapping, we propose the Temporal Response Averaging Algorithm which combines the temporal averaging step of the previous algorithm and unsupervised learning to produce data driven temporal contextual descriptors.

The third algorithm, the Temporal Contextual Trajectory Algorithm, maps a temporal sequence into an ordered trajectory representing the behavior over time of the input utterance on a 2-D map of emotions. The temporal information is integrated explicitly in the descriptor which makes it easier to monitor emotions in long speeches. The proposed mapping framework maps speech data of different length to the same equivalent representation which alleviates the problem of dealing with variable length temporal sequences. This is advantageous in real time setting where the size of the analysis window can be variable.

Another contribution of this thesis is the development of a novel data-driven emotion detection and recognition system that integrates the proposed framework. The goal of such system is to index speech databases to facilitate the classification and retrieval of emotions. The proposed system uses labelled emotional training data to estimate a learning model that is able to categorize unforseen speech data into one of the predefined emotions. The proposed system contains 4 main components. In the first step, preprocessing, the speech data is normalized with respect to the training data. In the feature extraction step, a set of acoustic features are extracted in order to reduce the size of the data for the subsequent steps while capturing the relevant emotional information. The proposed system works with any number of acoustic features. In the third step, the temporal feature representations of the speech data are mapped into global descriptors. The new descriptors are used in the subsequent steps of the system. In this thesis, the proposed framework is used as the feature mapping block of the system. In the final step, decision, the different descriptors of the speech data are independently classified then combined using score level fusion to produce a global decison/emotion for the input speech. In this thesis, we derive, use, and compare two score level fusion schemes.

In this work, we test the performances of the proposed system using two datasets.

The first, the Berlin Emotional Dataset, is an acted corpus. In this scenario, the emotional speech is acted by subjects in a professional manner. The actor is asked to speak a transcript with a predefined emotion.

The second dataset is an authentic dataset. In this scenario, the emotional speech is naturally recorded from spontaneous people in real life situations. Such situations include customer service calls, audio from video recordings in public places, and 911 calls. For that purpose, we have created and labelled a collection of debates. The proposed corpus is one of the first initiatives in the emotion analysis from speech literature.

The rest of this thesis is organized as follow. In the second chapter, we provide an extensive overview of the various steps and methodologies used in emotion analysis from speech. In the third chapter, we introduce the proposed framework and the proposed emotion detection and recognition system. In the fourth and fifth chapters, we evaluate the performances of the proposed work using two datasets. Chapter 6 highlights the contributions of this work and provides potential future research directions.

CHAPTER 2

RELATED WORK

In this chapter, we provide an overview of the various steps and methodologies used in emotion analysis from speech. After explaining definitions and modalities of emotion expressions, We start by outlining commonly used emotional datasets in the literature of emotion analysis from speech. Second, we review widely used preprocessing methods. Third, we outline standard acoustic features. Next, we review common feature mapping techniques used in speech analysis. Finally, we outline commonly used classifiers in the literature and validation techniques.

2.1 Emotion

2.1.1 Defining Emotion

Various definitions of emotion have been proposed in many fields such as psychology, psychiatry and, physiology. We researched definitions from various disciplines and we chose to adopt the following one:

In psychology, emotion is often defined as a complex state of feeling that results in physical and psychological changes that influence thought and behavior. Emotionality is associated with a range of psychological phenomena including temperament, personality, mood and motivation. About.com [14]. Last seen on 11/10/2014.

So how do we quantify these psychological changes? Emotions are personal. Research shows that humans can classify other people's emotions with an accuracy of 60% [15]. Emotions are expressed in different manners depending on factors such as sex, age, language, and culture. While yelling is a sign of anger and frustration in some cultures, in other cases, being loud is part of the culture itself. Language also plays a big role in emotions. For instance, the standard speech rate of Chinese language is faster than English. Moreover, speaking fast in English is usually a sign of worry and fear. People that aren't fluent in a specific language typically have a harder time to classify emotions. Consequently, teaching computers to detect emotion seems more challenging.

Emotions can be expressed in different modalities/sensors:

- Voice (speech): Humans communicate with each other mainly through speech. Typically, the speech conveys the intended message along with the emotional state of the speaker. Parameters such as tone, pitch, energy, and speech rate are strongly correlated with the emotional state of the speaker. In this modality, the emotional information is extracted from the speech signal of the input utterance.

- Images (Facial expressions): Facial expressions are also used by humans along with speech to convey, enrich, or emphasize any emotional state. Image is the most used modality in emotion analysis [1]. That is due to the extensive research in image processing and facial emotion analysis which led to the development of highly accurate systems for the detection and recognition of facial emotions [7, 8, 9]. In this case, the facial expressions of the speaker are analyzed using image processing and computer vision techniques in order to extract the emotional state of the speaker.

- Video (actions): Humans also convey emotional information using body gestures such as shaking head and/or hand. In this modality, the motion of the speaker is analyzed from the input video in order to extract the emotional state of the speaker.

- Physiological signals (EEG, EKG signals): Research has showed that the emotional state of humans is correlated with the EEG and EKG signals. In this case, the two signals are recorded and the emotional information of the speaker is extracted from the signals.

Choosing which modality to use to analyze emotions is a crucial decision. Due to many practical limitations, these modalities are rarely available at the same time. The choice of the appropriate modality depends mainly on the application and availability of the sensors. In some applications, the different modalities (when available) are combined in order to improve the analysis of emotions.

In other settings, the vocal information might be more accessible than images. In fact, nowadays, voice sensors are present in any mobile device. However, in contrast to facial emotions, the analysis of emotional speech has not met a wide success due to few challenges [5]. In this this, we focus on emotion analysis from speech using training computational models in order to tackle some of these challenges.

2.2 Emotion Analysis From Speech

A standard data driven speech emotion analysis system is decomposed into five main steps: 1) speech preprocessing, 2) feature extraction and selection, 3) feature mapping, 4) learning and classification, and 5) validation. An illustrative block diagram of a standard emotion analysis system is shown in figure 2.1. In this thesis, we mostly adopt this architecture.

The input to the system is a collection of annotated speech segments expressing various emotions.

First, preprocessing is applied in order to decrease the influence of speaker variability, background noise, and recording conditions on the subsequent steps of the system.

Second, a set of acoustic features is extracted from each audio segment. The goal of this step is to reduce the size of the data used for analysis. The chosen features must be representative of the underlying emotion. After feature extraction, some of the systems proposed in the literature go through a feature selection step [10]. The goal is to identify irrelevant features and eliminate their contribution in the decision making.

In the mapping step, the feature representation of each audio segment is mapped into a global descriptor which is representative of the entire speech. The new descriptor is used in the subsequent steps.

Next, the corpus is divided into training and testing sets. The training set is labeled and used by the classification scheme to estimate the parameters of a learning function that identifies the emotion of any given speech segment.

In the validation step, the testing set is used to evaluate and rate the performance of the classification scheme. In the next section, we review the literature of each step in details.

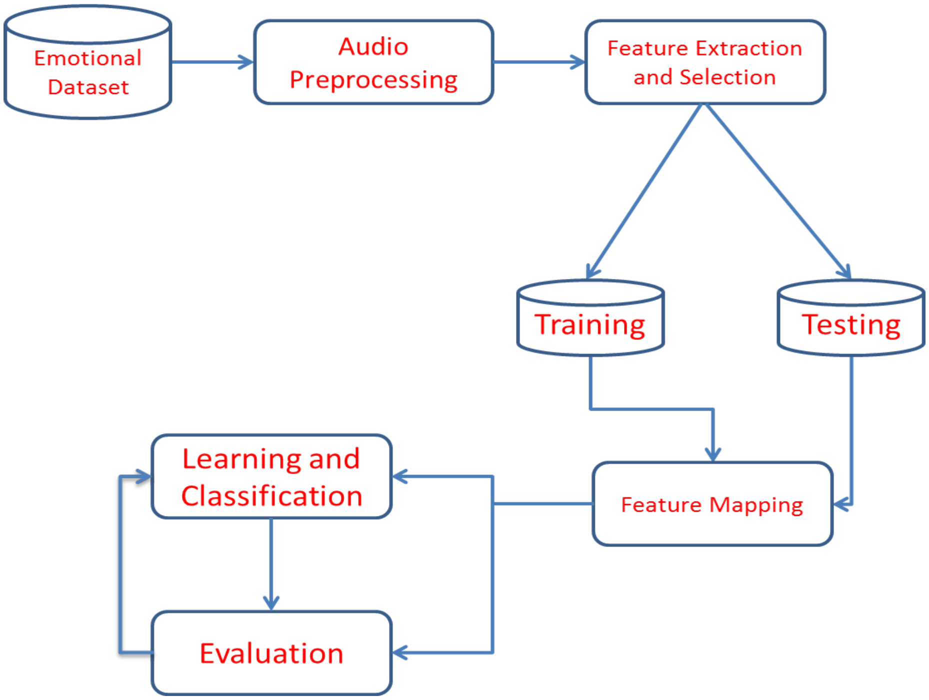


Figure 2.1: Architecture of a Standard Emotion Detection and Recognition System

2.2.1 Emotional Datasets

Emotion has a range of meanings, and it is crucial for the dataset to be clear where it stands in relation to the various emotional states. For instance, a person who develops dataset A might have a different point of view about emotions than a person who develops dataset B. The choice of an appropriate dataset for training computational models is fundamental. The data must be representative of the behaviors observed in the real application but also large enough, with sufficient variability of emotional expressions, including complex, mixed and shaded emotions. Likewise, expressions of emotions are best when collected as they occur in everyday actions and interactions rather than acted situations. Spontaneous emotions are hard to collect, to annotate, and to distribute due to privacy problems.

The proper preparation of an emotional speech database is a challenging task. There are various factors to be considered when selecting the dataset.

- Real vs. Acted emotions
- Number of emotions used in the dataset
- The parameters and the configuration of subjects such as number, gender (male, female, kids, seniors), and speech rate.
- Balanced vs. Unbalanced samples

Various datasets have been used in the literature[3, 16, 17, 18, 19, 2, 12, 11]. Most of the databases share the following emotions: anger, joy, sadness, surprise, boredom, disgust, and neutral [20]. There are 3 categories of datasets used in speech emotion analysis. These are acted, authentic or elicited emotional datasets.

- Acted Datasets: In this scenario, the emotional speech is acted by subjects in a professional manner. The actor is asked to speak a transcript with a predefined emotion. The Berlin Emotional Dataset [11] is an example of such corpus.

- Authentic Datasets: In this scenario, the emotional speech is naturally recorded from spontaneous people in real life situations. Such situations include customer service calls, audio from video recordings in public places, and 911 calls. The Vera-Am-Mittag (VAM) dataset [21] is an example of such corpus.

- Elicited Datasets: In this scenario, the emotional speech is induced with self-report instead of labeling. The emotions are provoked and experts label the utterances. The elicited speech is neither authentic nor acted. The Speech Under Stimulated and Actual Stress (SUSAS) dataset is an example of such corpus [3].

A good review of the three different types of datasets has been presented in [11]. Table 2.1 contains a list of commonly used datasets. At the early stages of emotion analysis, acted emotional datasets were the standard used in the literature. Gradually, the focus shifted towards more realistic datasets since acted ones simplify the problem of speech emotion analysis. Unfortunately, authentic datasets are harder to obtain due to privacy issues.

TABLE 2.1

A List of Common Datasets Used in Speech Emotion Analysis.

Name	Type	Number of Emotions	Configuration
Berlin Emotional Database	Acted	6	5 Males, 5 females
VAM	Authentic	5	19 Males, 13 females
SUSAS	Elicited	5	19 Males, 13 females

2.3 Notation

The starting point of any speech emotion analysis system is a collection of speech utterances. Let M denotes the number of unique emotions in the collection. Let $\mathbf{x}$ denotes the audio signal of any utterance in the corpus.

Typically, audio analysis is performed at the frame level. That is, the audio signal $\mathbf{x}$ is decomposed into overlapping frames where the frame size and overlap are fixed by the system (Typically, 10 to 100 ms duration is used as a frame size).

Frame level processing has two advantages. First, it leads to a more efficient process since the analysis is performed on short term audio signals. Second, the audio signal of the entire utterance is typically a non-stationary time series whose statistical properties depend on time. By using frame level analysis, the audio signal within each frame is stationary and the underlying information can be extracted more reliably.

Let $\mathbf{x}_i$ denotes the i^{th} frame extracted from $\mathbf{x}$ and N denotes the number of frames. Each utterance in the corpus have a variable length. Thus, N is variable from one utterance to another. For each frame $\mathbf{x}_i$, $1 \leq i \leq N$, a set of acoustic features is extracted. Let X_i, denotes the feature representation of $\mathbf{x}_i$ of dimension d. Thus, $\mathbf{x}$ is represented by a time series of feature row vectors $X = (X_{ij}, 1 \leq i \leq N, 1 \leq j \leq d)$. Figure 2.2 depicts the process of window decomposition of an input audio signal.

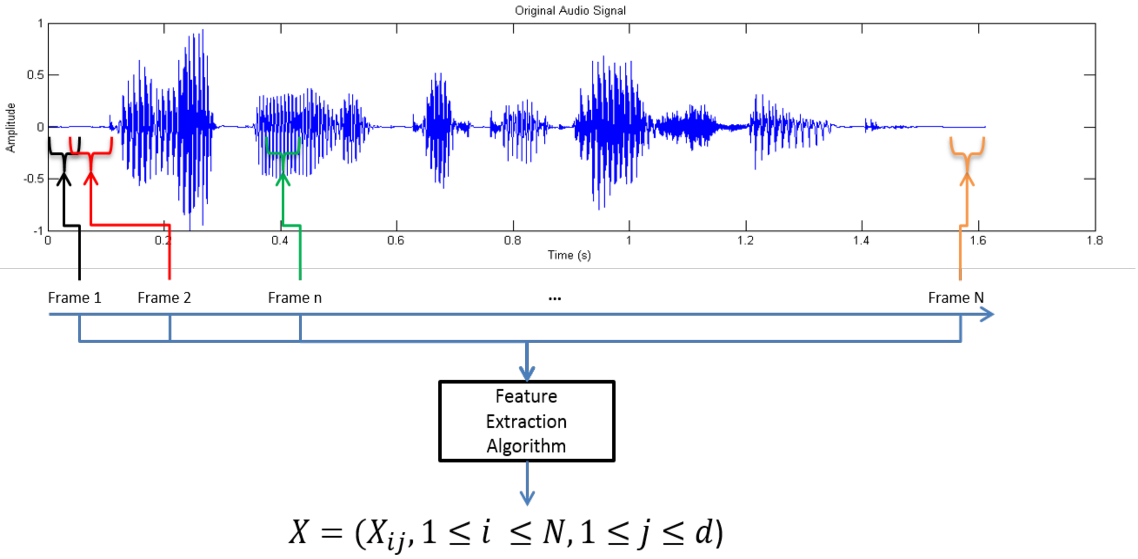


Figure 2.2: Window Decomposition of an Audio Signal.

2.4 Speech Preprocessing

Preprocessing is an important step in every speech analysis system.

First, corpses are typically biased by the variability of the different subjects and the recording conditions. Techniques such as speaker normalization [22] are used to decrease the influence of such factors.

Second, the audio signals are typically degraded by the background noise. Noise corrupts the signal and consequently deteriorates the performance of the system in the subsequent steps. Techniques such as spectral substraction [23] are used for noise reduction.

Various speech preprocessing techniques have been proposed in the literature [24]. In the next subsections, we present various preprocessing methods widely used in speech analysis.

2.4.1 Speaker Normalization

Speaker normalization is an acoustic preprocessing technique that reduces the influence of speaker variability. Ideally, emotion analysis systems should detect and recognize emotions regardless of the identity and characteristics of the underlying speaker.

In speaker normalization, the difference observed in the neutral subset of the dataset (the utterances labeled as neutral) is estimated then removed across all the speakers. The normalization parameters are then applied to the rest of the dataset (the utterances labeled as emotional). Thus, the discrimination between the different emotions is still preserved. This normalization is applied for each speaker separately. Two normalization techniques are widely used in speaker normalization.

Energy Normalization: The speech signals are scaled such that the average mean square energy of the neutral reference database across all speakers and the neutral subset in the emotional database are the same for each speaker [22]. Let E_{ref} and E^s_{ref} denote respectively the average root mean square energy of the neutral reference database across all speakers, and the average RMS energy of the neutral subset in the emotional database for speaker s. The scaling factor S^s_{energy} for the normalization is defined as

$$S^s_{energy} = \frac{E_{ref}}{E^s_{ref}} \tag{1}$$

The scaling factor S^s_{energy} in equation (1) is applied to each speaker s in the corpus. After normalization, the neutral samples of each speaker in the databases will have equal RMS value.

Given an audio signal $\mathbf{x}$ in the corpus, the new normalized signal $\mathbf{x}_{new}$ is computed as follow

$$\mathbf{x}_{new}(n) = \mathbf{x}(n) \times S^s_{energy} \tag{2}$$

The goal of this normalization is to reduce the influence of the different recording settings.

Pitch Normalization: Pitch is a widely used feature in emotion analysis [25, 26, 27]. The pitch is defined as the quality of a sound governed by the rate of vibrations producing it [28]. In other words, the degree of highness or lowness of a tone. Typically, pitch values have a high variability between different speakers. Consequently, the analysis in the subsequent steps are typically influenced by the pitch/identity of the underlying speaker. To overcome this issue, a pitch normalization can be applied [22]. That is, the pitch contour of each speaker in the database is normalized. Similarly to the previous normalization method, the average pitch across speakers in the neutral reference database ($F0_{ref}$) and the average pitch for each speaker ($F0^s_{ref}$) are estimated. We discuss and review various methods for pitch extraction in section 4.2.3.

Then, we compute and apply the scaling factor S^s_{pitch} defined as

$$S^s_{pitch} = \frac{F0_{ref}}{F0^s_{ref}} \tag{3}$$

S^s_{pitch} (In equation (3)) is used to scale the pitch contour of each speaker. After normalization, the neutral samples of each speaker in the databases will have equal pitch mean value. The goal of this normalization is to reduce the influence of the speaker identity.

By applying both normalization methods, we assume that the identities of the speakers in the dataset are known and the neural speech is available for each speaker. For real-case applications, this assumption is reasonable when either the speakers are known or a few seconds of their neutral speech can be prerecorded.

2.4.2 Signal Rectification

The Rectification of a signal consists in removing negative components. Two types of signal rectification are widely used namely full wave rectification and half wave rectification. Equations (4) and (5) represent respectively the full wave and half wave rectifications applied on an input signal $\mathbf{x}$.

$$\mathbf{x}_{fullwave}[n] = abs(\mathbf{x}[n]) \tag{4}$$

$$\mathbf{x}_{halfwave}[n] = \begin{cases} abs(\mathbf{x}[n]), & \text{if } \mathbf{x}[n] \geq \ 0 \\ 0, & \text{elsewhere} \end{cases} \tag{5}$$

Rectification is an essential step in many audio analysis tasks and feature extraction algorithms. Removing negative components is a crucial step in many feature extraction and speech analysis algorithms as it reduces the noise energy in the audio signal [29].

2.4.3 Spectral Substraction

The background noise is the most common factor degrading the quality and intelligibility of speech in recordings. Noise reduction techniques intend to lower the noise level without affecting the quality of the speech signal. Spectral substraction is a widely used method for noise reduction in speech analysis[23]. In this method, an average signal spectrum and an average noise spectrum are estimated from the audio signals and subtracted from each other, so that average signal-to-noise ratio (SNR) is improved. The mathematical equations and details of implementation of the spectral substraction algorithm are provided in appendix A.

2.5 Feature Extraction and Selection

After preprocessing, a set of acoustic features are extracted in order to reduce the size of the data for the subsequent steps. The chosen features must capture the different semantics of emotions. Choosing the right set of features to represent the different emotions is a crucial step for the success of the whole system. Typically, a set of acoustic features is used and combined in emotion analysis.

Different taxonomies exist for the categorization of acoustic features in the literature of emotion analysis. In [5], the authors classified audio features into three different categories namely: source features, vocal tract features, and prosodic features. In this thesis, we adopt this taxonomy. In the next subsections, we review these different categories of acoustic features.

2.5.1 Excitation Source Features

Source features are acoustic features extracted from excitation source signal [5]. The excitation source signal is extracted from speech by removing the vocal tract (VT) characteristics. The vocal tract is defined as the cavity in human beings and animals that is responsible for filtering sounds produced at the sound source (larynx in mammals; syrinx in birds). In order to remove the VT characteristics, first, linear prediction coefficients (LPCs) are used to estimate the VT information. Second, inverse filtering is used to remove the estimated VT information from the input signal. The resulting signal is the linear prediction (LP residual, and it contains mostly the information about the excitation source) [3, 30]. The details of computation of the LP residual signal are provided in appendix A. Features extracted from the LP residual are known as source features.

Various studies have been conducted on the use of excitation source features for speech analysis which showed that those features contains information such as message, speaker, language, and emotion information. In [6], the authors investigated the use of source features for emotion analysis. First, they ran a linear prediction analysis to estimate the LP residual which is also used to compute the Glottal volume velocity signal (GVV). The GVV contains important information about the excitation source. In the next step, they identified epochs which are defined as the instants of glottal closure where the signal to noise ratio of the signal is high. They proposed a set of features such as the sequence of LP residual samples and their phase information, parameters of epochs and their dynamics at syllable and utterance levels, and samples of GVV signal and its parameters. The authors used auto-associative neural networks (AANN) [31] and support vector machines (SVM) [32, 33] as classification scheme. Using two datasets namely the Telugu [34] and Berlin [3, 11, 12, 5, 35] emotion speech corpora, they showed that the combination of the different excitation sources features achieves competitive performances with other types of acoustic features used in the literature.

In [36], the authors showed that pitch information extracted from LP residual is efficient for speaker recognition. In [37], the authors used energy features extracted from LP residual for vowel and speaker recognition. Cepstral features derived from LP residual signal are used in [38] for speaker analysis.

Despite their good performances and wide use in different speech analysis tasks, excitation sources features are not widely used in the literature of emotion analysis via speech [6]. This is due to various reasons. First, it is due to the popularity of spectral features (Explained in the section 2.5.2). Second, emotion is present in the LP residual signal in the form of higher order relations and capturing those relations is still unsolved [6].

2.5.2 Vocal Tract Features

The extraction of vocal tract features is different from excitation source features. Vocal tract information are well defined in the frequency domain. Consequently, most of the VT features are computed based on fourier transform [39] and spectral analysis [40].

The extraction of vocal tract features follows these standard steps.

Framing: As discussed in the previous sections, the input audio signal is decomposed into overlapping small audio signals of length in the order of milliseconds. Short time segments are considered as stationary. Thus, standard signal processing techniques can be applied to capture the underlying information.

Spectral Analysis: For each frame in the input audio signal, spectral analysis techniques are applied to transform the signal from time to frequency domain. Statistics from the power spectrum are used as acoustic features. Examples include the spectral centroid, spectral roll-off, and spectral flux [29].

Sub-band Analysis: In order to extract more complex features, a sub-band analysis is performed by decomposing the power spectrum into sub-bands and applying feature extraction to each sub-band. Examples include the Mel Frequency Cepstrum Coefficients [35, 41, 42, 43].

Vocal tract features are widely used in emotion analysis. Mel Frequency Cepstrum Coefficients (MFCCs), Linear Predictive Cepstral Coefficients (LPCCs), Perceptual Linear Prediction Coefficients (PLPCs), and formant features are the most used vocal tract features [35].

In [41], the authors proposed an emotion recognition system based on MFCCs, LPCCs, RASTA PLP coefficients and LFPCs. They tested their approach using a Mandarin emotional dataset and showed that vocal tract features are efficient for emotion analysis. In [42], the authors used log frequency power coefficients and a four stage ergodic hidden markov model (HMM) to classify emotions into six classes. In [43], the authors reported that MFCC outperform pitch features in emotion recognition.

2.5.3 Prosodic Features

Prosody is responsible for structuring the flow of speech. Duration, tone, and intensity are important characteristics in the production of speech and makes human speech more natural. The prosody is represented acoustically by the patterns of duration, tone, and energy. Consequently, prosodic features represent the perceptual speech properties. Human emotional expressiveness (i.e. emotionally excited behavior of articulators) can be captured through prosodic features. The prosody is manifested at four different levels namely: Linguistic intention, articulatory, acoustic realization, and perceptual level.

- Linguistic level: Prosody is the rhythm, stress, and intonation of speech. For example, a question usually have a different tone than a statement.

- Articulatory level: Prosody is manifested as a series of articulatory movements.

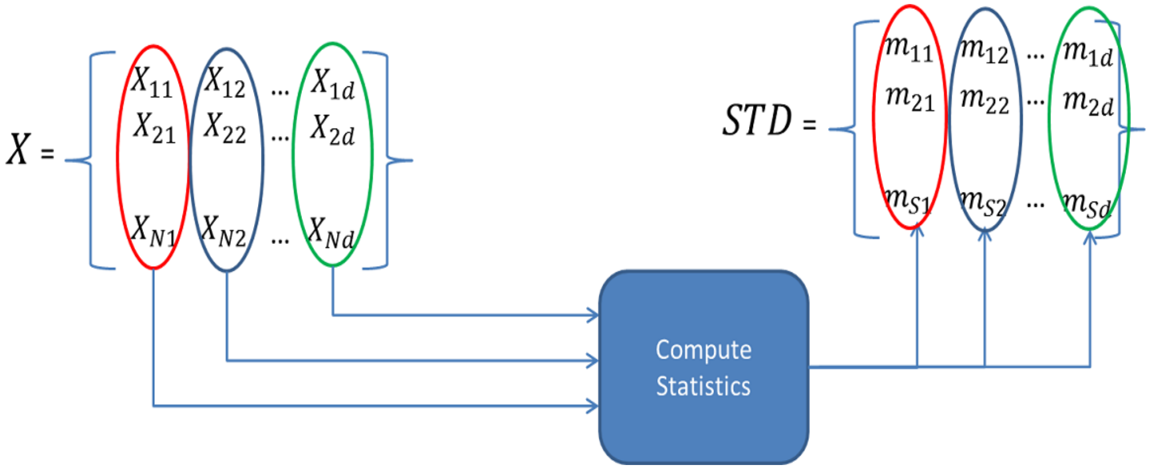
- Acoustic level: Prosody can be quantified using analysis of acoustic parameters such as fundamental frequency (F0), intensity, and duration. For instance, a stressed speaker will have higher fundamental frequency, greater amplitude, and longer duration than an unstressed speaker.

- Perceptual level: Prosody is quantified through perceptual processing from the listener via subjective experience such as pauses, length, melody and loudness of the perceived speech.

Pitch, energy, duration, and their derivatives are the most used prosodic features in emotion analysis[3]. Statistics computed from these features are used to discriminate between emotion states [16, 17]. In particular, pitch and intensity seem to be correlated to the amount of energy required to express a certain emotion [20]. In [19], the authors proposed an approach for emotion recognition. Initially, they started with 86 prosodic features. After feature selection, they chose the top 6 features. They reported 92% accuracy on a basque emotional database. Energy, pitch and duration based features are used in [18] to classify 6 emotions in Mandarin language. The authors reported 88% accuracy using SVM and genetic algorithms.

2.6 Feature Mapping

After feature extraction, the feature representation of each utterance is mapped into a global descriptor which is representative of the entire utterance. The new descriptor is used in the subsequent steps of the system. The most used mapping approach in emotion analysis computes statistics over the entire sequence and use it as a global descriptor [10, 20, 15, 3, 4]. The resulting descriptor is used for analysis, learning, and classification. We refer to this approach by the Statistics Based Mapping Algorithm (SBMA). As discussed in the previous section, acoustic features are extracted from short term analysis windows. Thus, an audio utterance is represented by a multivariate time series. The SBMA works as follow: Given a sequence of features $X = (X_{ij}, 1 \leq i \leq N, 1 \leq j \leq d)$ of N rows of temporal data with d columns. For each feature j where $\{1 \leq j \leq d\}$, S statistics such as min, max, mean, median, and standard deviation are computed over the whole sequence resulting in a $S \times d$ representation. A global feature vector is formed by concatenating the statistics matrix into a vector resulting into a $1 \times (S \times d)$ representation. Figure 2.3 represents the different steps of the Statistics Based Mapping Algorithm.



X =
X_{11} X_{12} ... X_{1d}
X_{21} X_{22} ... X_{2d}
X_{N1} X_{N2} ... X_{Nd}
Compute Statistics
STD =
m_{11} m_{12} ... m_{1d}
m_{21} m_{22} ... m_{2d}
m_{S1} m_{S2} ... m_{Sd}

Figure 2.3: Architecture of the Statistics Based Mapping Algorithm (SBMA) .

2.7 Classification and Learning

After feature extraction and mapping, the corpus is split into training and testing sets. The training set is used by a classifier to learn a set of classification rules. A classifier is a function f that maps an input feature representation X to a predefined emotion. In our case, the classification function take as an input the descriptor of a speech segment and it returns the predicted emotion of the speaker. Classifiers can be categorized into linear and non-linear. In the next sections, we will review the literature of both categories.

2.7.1 Linear Classifiers

A linear classifier bases its classification decision on a linear combination of input features. Let $Y = \{Y_i, 1 \leq i \leq N\}$ denotes an input feature vector of dimension d. The output score of the classifier is given by

$$y = f(w.Y) = f(\sum_{j=1}^{d} w_j.Y_j) \tag{6}$$

In (6), w denotes the weight vector which is estimated from the training data. f denotes the classification function, Y_j denotes the j^{th} value of Y. Linear classifiers compute a linear separator between classes in the feature space.

Various linear classifiers have been proposed in the literature [44, 45]. Examples of linear classifiers are the Naive Bayes Classifier [46], Perceptron classifiers [47].

Linear classifiers are usually governed by few parameters. These parameters can be estimated using 2 different approaches namely generative and discriminative. The first approach uses probability density functions. Examples of such classifiers are the Naive Bayes classifier, linear discriminant analysis, Fischer's linear discriminant analysis [48]. On the other hand, the second types of classifiers adopt a non probabilistic approach but instead works on discriminative properties of the features. Examples of such classifiers are Perceptron classifiers [47], linear support vector machines [32, 33].

2.7.2 Non-Linear Classifiers

Linear classifiers are straight-forward however they are not efficient in real cases since data usually discontinue the class borders. To overcome this issue, a non-linear transformation is first applied to the input feature vectors. The goal of this step is to project the data into a new feature space where the problem is linearly separable. The second step is to compute the linear boundaries in the new feature space.

These two steps constitute the main idea of non-linear classifiers. Non-linear classifiers are defined as follow

$$y = f(w.h(Y)) = f(\sum_{j=1}^{d} w_j.h(Y_j)) \tag{7}$$

In (7), w denotes the weight vector which is estimated from the training data. f denotes the classification function. Y_j denotes the j^{th} value of Y. h is the transformation function.

Various non-linear classifiers have been proposed in the literature [45]. Examples of non-linear classifiers are Hidden Markov model [49, 50], Gaussian mixture models (GMM) [51, 52], non-linear support vector machines [32, 33], neural networks [31], and decision trees [53].

Non-linear classifiers are more complex to implement which results into a higher computational complexity. Despite that, they are more used in emotion analysis literature than linear classifiers due to the non-linear property of speech data.

2.8 Fusion

The fusion of different features or classifiers improves the overall performances of any system. This is due to two main reasons.

As discussed in section 2.5, despite the large number of proposed features in the literature, emotion is still a complex notion that depends on several factors and there is no feature that represent all the aspects of emotion. Consequently, most of the proposed methods in the literature combine features from the three categories to improve the representation, detection, and recognition of human emotions [35, 20, 19, 18, 43, 16]. This is also the case for other disciplines such as character recognition, speech recognition, etc.

As discussed in section 2.7, many classifiers have been proposed in the literature. These classifiers are based on different theories and it was shown that their performances overlap. Thus, the fusion of classifiers with different structures and configurations improves the overall performance of the system.

Different taxonomies exist for Fusion techniques [54]. Typically, these fusion techniques are divided into four different categories namely: data level, feature level, score or soft level, and decision or hard level fusion. These levels are closely related to the abstraction level of the data throughout the system. Data level fusion is rarely used in emotion and speech analysis. Thus, we do not discuss it further in this thesis.

In the next subsections, we review the other three categories.

2.8.1 Feature Level Fusion

In feature level fusion, the different acoustic features are typically combined to form a global feature vector by applying appropriate feature normalization, transformation, and concatenation.

The resulting global feature vector is fed to a classifier for learning and classification. The advantage of applying feature level fusion is the decrease of computational complexity. Instead of using F classifiers, the F acoustic features are first combined into one feature set and one single classifier is used for learning. A simple and widely used feature level fusion approach is to concatenate the different acoustic features into one single feature vector[55, 56]. In [55], the authors applied simple concatenation to combine acoustic features and improve the overall classification performance. The disadvantage of feature level fusion is that the system do not profit from the strengths of the individual features.

In [57], the authors performed a comparison between feature level and decision level fusion. The authors extracted acoustic and linguistic features. The authors applied decision level fusion by combining the decision of the acoustic and linguistic classifiers using decision trees. Then, they applied feature-level fusion by concatenating both sets of features before learning. They showed that the results achieved by the classifiers using the parameters merged at feature level outperformed the classification results of the decision-level fusion scheme.

2.8.2 Score Level Fusion

In score level fusion, each classifier computes a score vector for each data sample. The score vector of length M represents the confidence degree that the input sample belong to each of the M classes. The score vectors output by the different classifiers are combined in order to compute a global score vector.

Typically, statistics such as max, mean, or median computed from the score vectors of the different classifiers are used as the global vector. Score level fusion is the most common category due to the ease of accessing scores output by classifiers. Moreover, score level fusion can applied to classifiers that are density, neighborhood or distance based such as K-NN [54]. Typically, a transformation is first applied on these classifiers to transform their outputs into a score representation. Then, score level fusion is applied. An example of score level fusion methods is the Bayesian Fusion. In [58, 59], the authors introduced two statistical frameworks for score level fusion of classifiers based on the Bayesian theorem. The first framework proposed a fusion method for classifiers that use distinct representation in the input layer. In the second framework, they proposed a fusion scheme for classifiers that have a shared representation. Finally, they showed that the two frameworks can also be used for the fusion of classifiers that have a subset of the input representation as distinct and the other subset of the input as shared between the classifiers.

In [54], the authors introduced an approach for the fusion of Bayes classifiers that could be also extended to other types of classifiers such as K-NN and Neural Networks.

2.8.3 Decision Level Fusion

Decision level fusion can be considered as a higher abstraction of score level fusion. In score level fusion, the classifier outputs a score vector for each input sample. In decision level fusion, the score vector of an input sample is mapped to a final label. Thus, each classifier attributes a final label or decision for the input utterance. Then, the decision of the different classifiers are combined in order to compute a final decision label. A simple and widely used approach is to choose the majority class among the different classifiers [54, 57, 60]. Other involved approaches the Borda Count [61] or simply use the different decisions by the classifiers to form a new feature vector that is fed to a classifier for learning and final decision. A detailed review of decision level fusion techniques can be found in [62].

2.9 Validation

2.9.1 Performance Metrics

As discussed in the previous sections, various classification schemes and datasets have been proposed in the emotion analysis literature. First, we need a set of appropriate measures of performance that permit easy analysis of the performance of the proposed system and how it compares across datasets and other state of the art techniques.

Measures of performance must satisfy several criteria in order to make the analysis logical. These criteria are

- Coherence: The performance measures need to capture the aspect of performance of interest.
- Maturity: The performance measures need to be widely used by the community so it provides a way to analyze and compare the proposed system against state

of the art methods from the literature.

- Computational complexity: The performance measures must be computationally inexpensive so they can be applied on big datasets.

Various studies have been conducted on performance measures for supervised learning algorithms [63]. In the next subsections, we review commonly used performance measures in the literature.

2.9.1.1 Misclassification Counts

It is the standard performance measure of a classifier. The predicted labels of the validation or testing data are compared its true label. Two statistics namely false negative (FN) and false positive (FP) can be derived. The FN and FP and their complements, namely true positive (TP) and true negative (TN), can be used to form the confusion matrix and error rate ($ER = \frac{FN+FP}{FN+FP+TP+TN}$) defined also as the trace of the confusion matrix divided by the number of examples.

2.9.1.2 ROC Curves

Classifiers can trade-off one type of misclassification for another. This trade-off is typically represented by the receiver operation characteristic (ROC) curve. In the ROC curve, the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points of a parameter. Thus, it provides the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between classes. Figure 2.4 represents the comparison of the ROC curves of two classifiers. From the figure, we can conclude that the TCT and TRAA classifiers outperform the TAA classier. Moreover, the performances of the TRAA and TCT are correlated.

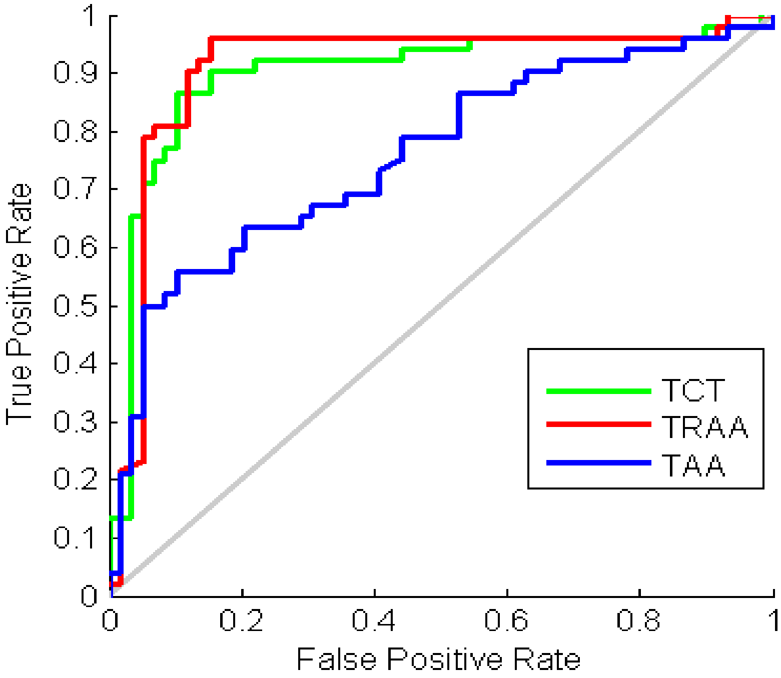


Figure 2.4: Comparison of the ROC curves of three classifiers

CHAPTER 3

EMOTION ANALYSIS USING TEMPORAL CONTEXTUAL DESCRIPTORS

In this chapter, we introduce our feature mapping framework. First, we discuss the motivation behind our approach. Second, we outline the proposed framework. Finally, we use the proposed feature mapping framework to build a speech emotion detection and recognition system that is able to index and classify human emotions from speech.

3.1 Motivations

The current trends in technology suggest that the next generation of services and devices allows smarter customization and automatic context recognition. Computers learn the behavior of the users and can offer them customized services depending on the context, location, and preferences.

Emotion analysis systems must be practical for real time services which will enable them to be used in many contextual applications such as SIRI where an emotion recognition system will be able to provide a deeper connection between the human and the phone based on the emotional state of the user. It can also be used in call centers where calls are logged and emotion analysis can be used to detect and monitor negative emotions from clients. Emotion analysis can also be used to monitor emotions in debates in order to detect confident/confused speakers.

Ideally, emotion detection and recognition systems should work simultaneously with speech recognition systems to recognize speech and emotion at the same time. That will lead to develop smart speech recognizers that are emotion dependent.

Most of the methods proposed in the literature are focused on accuracy rather than practicality. One of the long-standing criticisms of academic research is that the ideas and techniques developed are simply not practical for real services which can be embedded in tiny devices and works in real time. There are few criteria that must be considered before developing emotion detection and recognition systems. These criteria are computational complexity, simplicity of the solution, and efficiency. As discussed in the previous chapter, acoustic features are extracted from short term analysis windows. Thus, an audio segment is represented by an ordered (in time) sequence of features or a multivariate time series. Typically, the sequence is further processed to compute a global descriptor representative of the entire utterance/sequence. In the previous chapter, we introduced the Statistics Based Mapping Algorithm (SBMA), a widely used mapping algorithms in the literature of speech emotion analysis.

The Statistics Based Mapping Algorithm has various limitations.

First, the mapping of X to a global descriptor results in the loss of temporal ordering of the original sequence. Emotion is considered as a gradual acoustic process or a succession of acoustic events. By discarding the temporal ordering in the mapping, the Statistics Based Mapping Algorithm cannot detect acoustic patterns that lead to a certain emotion. Feature mapping algorithms must integrate the temporal ordering of sequential data in order to improve the representation of emotion descriptors. Preserving the temporal ordering proved to be efficient in other speech analysis tasks such as speech recognition and emotion analysis [64, 13, 65, 56, 66]. This also the case for other types of time series data. For instance, the temporal ordering in financial time series data is important in the prediction of the future prices. In fact, the price fluctuations in the stock market are strongly correlated with the temporal activity of the prices.

Second, the Statistics Based Mapping Algorithm is not practical for real time emotion analysis. In real time settings, segments are rather a continuous stream of speech and emotion should be analyzed continuously. The Statistics Based Mapping Algorithm assumes that the input sequence contains only one emotion. The main challenge of using The Statistics Based Mapping Algorithm in real time settings is the choice of the segment's length. If the segment is too short, we risk not having enough information to analyze. If the segment is too long, we risk having two or more emotions in the same segment. In other words, the Statistics Based Mapping Algorithm cannot detect changes in emotion in long utterances.

Third, the Statistics Based Mapping Algorithm derives each utterance independently. In [56, 66, 13, 65], the authors showed that data driven feature mapping algorithms are more efficient in the discrimination between classes.

Motivated by these issues, we introduce a feature mapping framework. The proposed framework maps sequential data into global descriptors that integrate the temporal ordering of the original sequence. The proposed framework uses unsupervised learning to compute data driven descriptors that can be adapted to real time emotion analysis. We use the proposed feature mapping framework to build a speech emotion detection and recognition system. In the next sections, we provide an overview of the proposed framework and emotion detection and recognition system.

3.2 Overview of The Proposed Feature Mapping Framework

The proposed framework maps a temporal sequence of information into temporal contextual descriptors. In fact, it can be generalized to any other type of ordering such as spatial or frequency ordering. It can also be used with other types of time series data as long as the assumptions of the proposed framework are met. The proposed framework decreases the computational complexity by mapping a temporal sequence of information into a reduced representation while preserving the temporal information from the original features. In this thesis, we propose three algorithms for integrating the temporal ordering in the mapping.

In the first algorithm, the Temporal Averaging Algorithm, the data is averaged using leaky integrators [13] to produce a global descriptor that implicitly integrates the temporal information of the original sequence.

In order to integrate the discrimination between classes in the mapping, we propose the Temporal Response Averaging Algorithm which combines the temporal averaging step of the previous algorithm and unsupervised learning to produce data driven temporal contextual descriptors.

The third algorithm, the Temporal Contextual Trajectory Algorithm, maps a temporal sequence into an ordered trajectory representing the behavior over time of the input utterance on a 2-D map of emotions. The temporal information is integrated explicitly in the descriptor which makes it easier to monitor emotions in long speeches.

3.2.1 Notations

Recalling the notations used in section 2.3, an utterance $\mathbf{x}$ is represented by its feature representation $X = (X_{ij}, 1 \leq i \leq N, 1 \leq j \leq d)$. X represents a sequence of features ordered in time. In this work, our goal is to integrate the temporal ordering in the feature representation and map features into temporal contextual global descriptors and investigate their performances.

The mapping of X into a temporal contextual descriptor can be considered as a function MAP defined as follow

$$\Re^{N\times d} \rightarrow \Re^{E} \tag{8}$$

$$MAP(X) \rightarrow \mathbf{Y} \tag{9}$$

In (9), $\mathbf{Y}$ denotes the output of mapping X using function MAP. In the next sections, we introduce three different mapping algorithms.

3.2.2 Temporal Averaging Algorithm

The Temporal Averaging Algorithm consists of a simple averaging of the input sequence over time. In order to average the sequence, we use leaky integrators [13]. A leaky integrator is a differential equation that is used to describe a system that takes the integral of an input and gradually leaks a small amount of information over time. The concept of leaky integrators is inspired from electrical circuit theory where the voltage of a capacitor would be the integral of the current if the capacitor did not leak electrons.

In our case, the leaky integrator is used to average the sequence overtime while leaking some information from the past samples in the computation of newer samples. We researched leaky integrator functions [67, 13, 68] and we decided to use the function introduced in [13]. The averaging is a straightforward step and can be performed using the following equation:

$$\mathbf{Y}_{i+1} = w.\mathbf{X}_i + (1-w).\mathbf{Y}_i \tag{10}$$

In (10), $\mathbf{X}_i$ denotes the i^{th} feature vector in X. $\mathbf{Y}_i$ is the i^{th} output vector. w denotes the rate of leakage or the weighting parameter which controls the size of the memory associated with past samples. The greater the value of w, the lesser the previous patterns will contribute in the averaging computation.

After averaging the entire sequence, the vector $\mathbf{Y}_{N+1}$ contains temporal information about the entire sequence since all data samples contributed to its computation.

Thus, the mapping of X using the Temporal Averaging Algorithm is defined as follows

$$\Re^{N\times d} \;\rightarrow\; \Re^{d} \tag{11}$$

$$MAP(X) \;\rightarrow\; \mathbf{Y}_{N+1} \tag{12}$$

In (12), $\mathbf{Y}_{N+1}$ denotes the temporal average of the entire sequence computed using equation 10. In the rest of this dissertation, we refer to the mapping of X using the Temporal Averaging Algorithm by $TAA(X)$. Figure 3.5 represents the different steps of this approach.

Figure 3.6 represents the comparison of the Temporal Averaging Algorithm Descriptors of a speech utterance spoken in neutral and affective states respectively. The audio signals of the two utterances are decomposed into overlapping frames. The frame size is fixed to 40 milliseconds with 50% overlap (That is 20 ms for the hop size). For each frame, we extract 13 MFCC coefficients. Thus, each utterance is represented by a sequence of MFCC features. Next, we map the sequences of the two utterances using the Temporal Averaging Algorithm as described in the previous paragraph. The memory parameter is set to $w = 0.05$. After mapping using the Temporal Averaging Algorithm, the utterance is mapped to a 1×13 descriptor. As it can be seen, the Temporal Averaging Descriptors have a different behavior for each emotion. We can notice that the neutral and affective segments have similar values at some MFCC coefficients such as coefficient 1,2,3 and 10 while they have different values at some other MFCC coefficients 4 to 8. This suggests that the emotional information is contained in coefficients 4 to 8. In fact, the MFCC coefficients represent the power of short term spectrum computed on the mel-scale frequency. In our case, we computed 13 MFCC coefficients. That means that each MFCC coefficient represent the power spectrum at a specific frequency range. Different emotions are manifested at different frequency ranges which explains the different in MFCC coefficients 4 to

8 in the two segments.

The Temporal Averaging Algorithm maps sequences of different length to the same equivalent representation which alleviates the problem of dealing with variable length temporal sequences. It also makes the fusion at the feature level easier since each feature set is mapped to one feature vector. One possible scenario for fusion is the concatenation of the different feature sets after mapping. The Temporal Averaging Algorithm is also computationally efficient since the subsequent steps of the systems will be performed on $\mathbf{Y}_{N+1}$ instead of the sequence X.

The Temporal Averaging Algorithm integrates the temporal ordering implicitly. However, despite its simplicity, the Temporal Averaging Algorithm does not take into account the discrimination between emotions. That is, the Temporal Averaging Descriptors are not data driven. It is also ambiguous for real time emotion analysis since it integrates the temporal information implicitly.

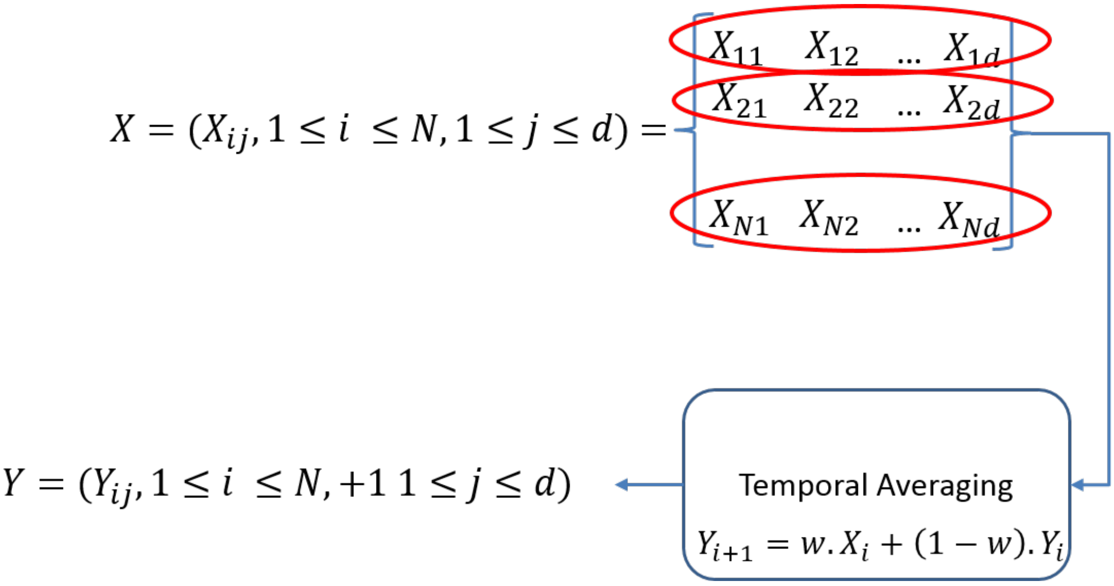

$X = (X_{ij}, 1 \leq i \leq N, 1 \leq j \leq d) =$
X_{11} X_{12} ... X_{1d}
X_{21} X_{22} ... X_{2d}
X_{N1} X_{N2} ... X_{Nd}
Temporal Averaging
$Y_{i+1} = w.X_i + (1-w).Y_i$
$Y = (Y_{ij}, 1 \leq i \leq N, +1\ 1 \leq j \leq d)$

Figure 3.5: Architecture of The Temporal Averaging Algorithm.

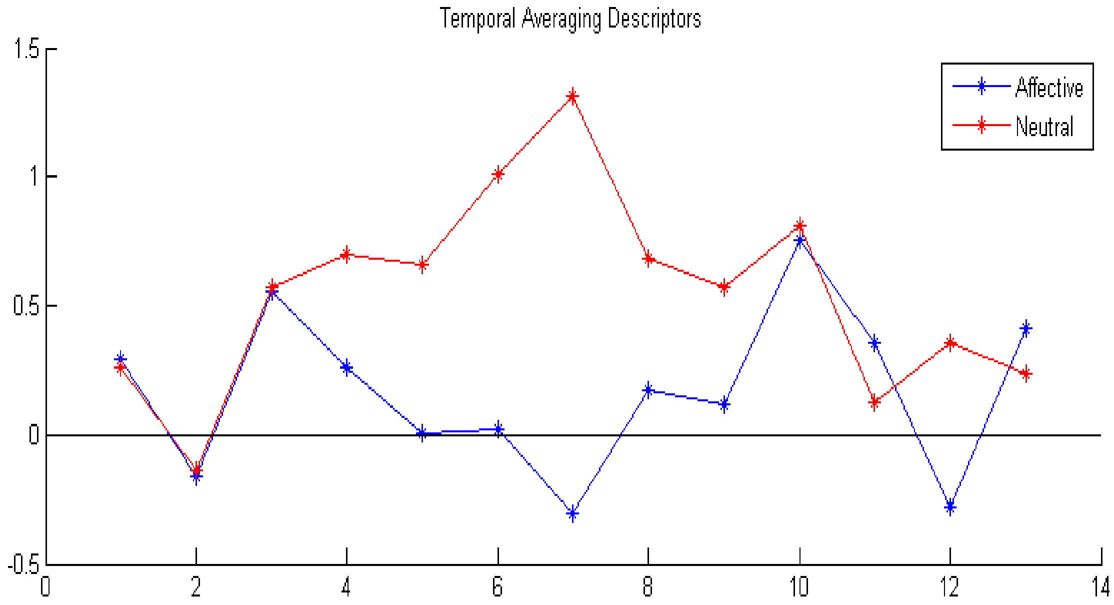


Figure 3.6: Comparison of the Temporal Averaging Algorithm Descriptors of a Speech Utterance Expressed in Neutral and Affective (Anger) State Using the MFCC Features.

3.2.3 Temporal Response Averaging Algorithm

The Temporal Response Averaging Algorithm is an extension of the Temporal Averaging Algorithm. In the latter approach, we averaged the sequence over time to produce a temporal contextual descriptor using leaky integrators. In the Temporal Response Averaging Algorithm, each vector in the sequence is mapped using SOMs then their responses are averaged overtime using leaky integrators [13]. The basic SOMs algorithm is provided in appendix A. Let $\mathbf{X}_i$ denotes the i^{th} feature vector in X. Let $\mathbf{Y}_i$ denotes the activity distribution of $\mathbf{X}_i$ on a trained SOMs. In this case, the training data can be used to train the SOMs. After mapping, the sequence X is represented by $Y = (\mathbf{Y}_i, 1 \leq i \leq N)$. Y is a sequence representing the response of X on the SOMs. Y is averaged over time to produce a new output Z using leaky integrators:

$$\mathbf{Z}_{i+1} = w.\mathbf{Y}_i + (1 - w).\mathbf{Z}_i \tag{13}$$

In (13), $\mathbf{Z}_i$ denotes the i^{th} output vector. w denotes the rate of the leak or the weighting parameter which controls the size of the memory associated with past samples. The greater w, the more previous patterns will be forgotten in the averaging computation.

The vector $\mathbf{Z}_{N+1}$ contains temporal information about the entire sequence since all data samples contributed to its computation.

Thus, the mapping of X using the Temporal Response Averaging Algorithm is defined as follow:

$$\Re^{N\times d} \;\rightarrow\; \Re^{s^2} \tag{14}$$

$$MAP(X) \;\rightarrow\; \mathbf{Z}_{N+1} \tag{15}$$

In (15), $\mathbf{Z}_{(N+1)}$ is computed using equation (13). s denotes the size of the SOMs. In the rest of the thesis, we refer to the mapping of X using the Temporal Response Averaging Algorithm by $TRAA(X)$. Figure 3.7 displays the different steps of the Temporal Response Averaging Algorithm.

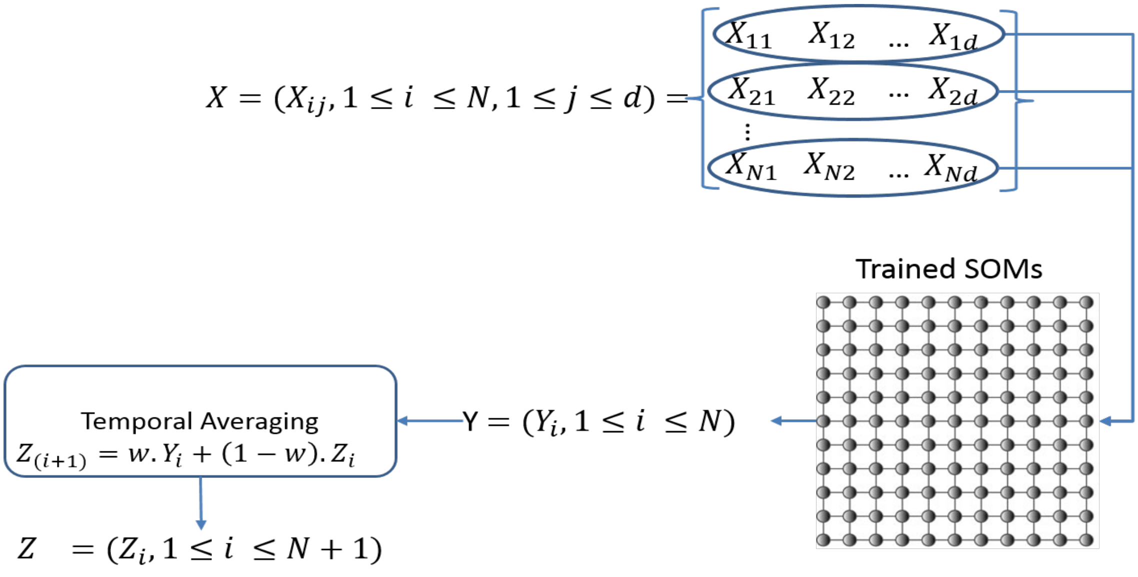
The computation in equation (15) is thus not only based on the current data sample, but also on the time average of the responses of previous patterns induced from the map. The purpose of applying SOMs is to distinguish between different areas in the feature space which are represented in the output space by different dimensions (Best Matching Units Activity). The response of each feature vector on the map provides insights on which region of the map the input vector belongs to. Although the responses are averaged over time, the dimensionality (which captures the different areas in the feature space) is saved. Thus, the Temporal Response Averaging Algorithm is expected to integrate more efficiently the temporal information and achieve better discrimination power than the Temporal Averaging Algorithm.

Figure 3.8 represents the comparison of the Temporal Response Averaging Algorithm

Descriptors of the same example in figure 3.6. We used the MFCC features with 13 coefficients. The frame size is set to 40 milliseconds with 50% overlap. The memory parameter is set to $w = 0.05$. We used a 7×7 map size for the SOMs. Thus, the Temporal Response Averaging Algorithm maps the sequence of MFCC features into one feature vector of dimension 49. The response on the SOMS can be computed in three different ways namely crisp, fuzzy, and kernel. In this example, we used a crisp response.

As it can be seen, the Temporal Response Averaging Descriptors of the two utterances have different distributions on the map units. For instance, the neutral utterance have a high response to other map units where the affective utterance do not have any activity and vice versa. Since the Temporal Response Averaging Algorithm is data driven, it is expected to achieve better discrimination between the different emotions than the Temporal Averaging Algorithm.

The Temporal Response Averaging Algorithm integrates the temporal ordering implicitly and takes into account the discrimination between classes. However, the algorithm is still ambiguous since it integrates the temporal information implicitly.



$X = (X_{ij}, 1 \le i \le N, 1 \le j \le d) =$
X_{11} X_{12} ... X_{1d}
X_{21} X_{22} ... X_{2d}
⋮
X_{N1} X_{N2} ... X_{Nd}
Trained SOMs
$Y = (Y_i, 1 \le i \le N)$
Temporal Averaging
$Z_{(i+1)} = w.Y_i + (1-w).Z_i$
$Z = (Z_i, 1 \le i \le N+1)$

Figure 3.7: Architecture of The Temporal Response Averaging Algorithm.

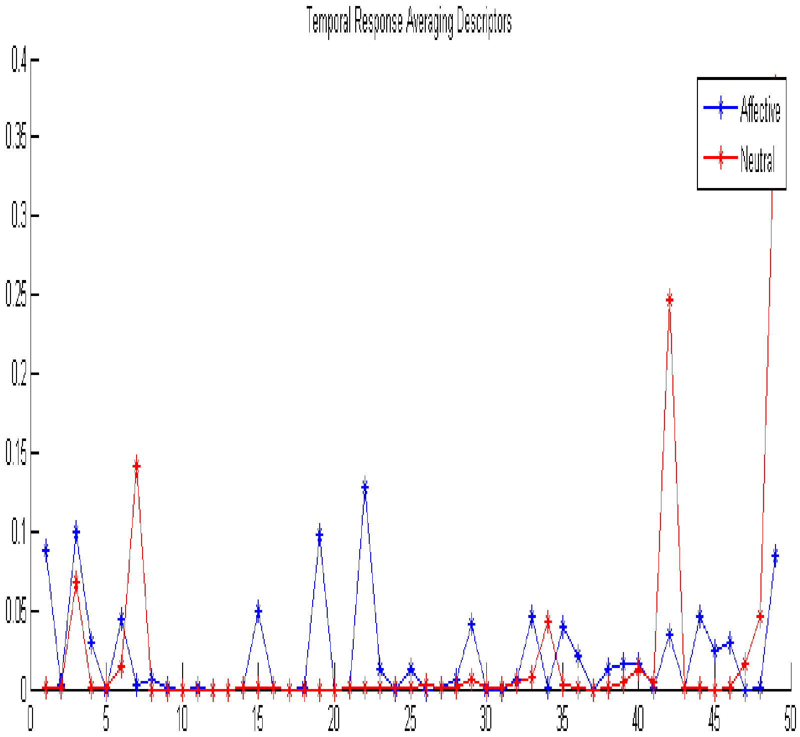


Figure 3.8: Comparison of the Temporal Response Averaging Algorithm Descriptors of a Speech Utterance Expressed in Neutral and Affective (Anger) State Using the MFCC Features.

3.2.4 Temporal Contextual Trajectory Algorithm

As we pointed out in the previous sections, the Temporal Averaging Algorithm and the Temporal Response Averaging Algorithm integrates the temporal information implicitly. Ideally, the temporal ordering should be integrated explicitly in the representation. In this section, we introduce the Temporal Contextual Trajectory Algorithm which uses the topology preserving of SOMs and the continuous nature of speech to derive data driven descriptors that integrate the temporal ordering explicitly and suitable for real time emotion analysis.

Due to topology preserving properties of SOMs, two adjacent vectors on the map are physically alike [69]. On the other hand, due to the continuous nature of speech, two consecutive frames are also physically alike. Consequently, their feature representations are also similar. During clustering by SOMs, these consecutive feature vectors are mapped to similar best matching map units on the SOMs. Thus, the sequence of feature vectors is mapped to an ordered trajectory on the map by SOMs. Each point in the trajectory represents the coordinates of the best matching unit of the corresponding feature vector.

Let X denotes the input feature sequence to the SOMs. Let $\mathbf{Y}_i$ denotes the best matching unit of the i^{th} feature vector $\mathbf{X}_i$. The feature sequence $X = (\mathbf{X}_i, 1 \leq i \leq N)$ is represented by a trajectory of length N. Let $Y = ((y_{i1}, y_{i2}), 1 \leq i \leq N)$ denotes the 2-D trajectory of X on the map.

Figure 3.9 represents the trajectory of a speech segment spoken in neutral state on a 2-D SOMs. The audio signal is decomposed into overlapping frames. The frame size is fixed to 40 milliseconds with 50% overlap (That is 20 ms for the hop size). For each frame, we extract 13 MFCC coefficients. Thus, each segment is represented by a sequence of MFCC features. Next, we map the sequences of the two utterances into Y as described in this paragraph. We used a map size of 7×7 for the SOMs. Since the trajectory is ordered in time, we refer to each position i in the trajectory by epoch i. As it can be seen, the trajectory in figure 3.9 is often long, redundant, and noisy. In fact, an audio segment of 2 seconds is represented by more than 100 frames if the frame and hop size are set to $40ms$ and $20ms$ respectively. Thus, the length of its trajectory on the SOMs is 100. On a 7×7 map size, a trajectory of length 100 is ambiguous for analysis.

The redundancy is due to the topology preserving property of the SOMs. Since consecutive frames are mapped into similar map units and often on the same map unit, the trajectory suffers from redundancy.

The noise is due to noisy frames which are mapped on random parts of the map and thus increase the variability of the trajectory.

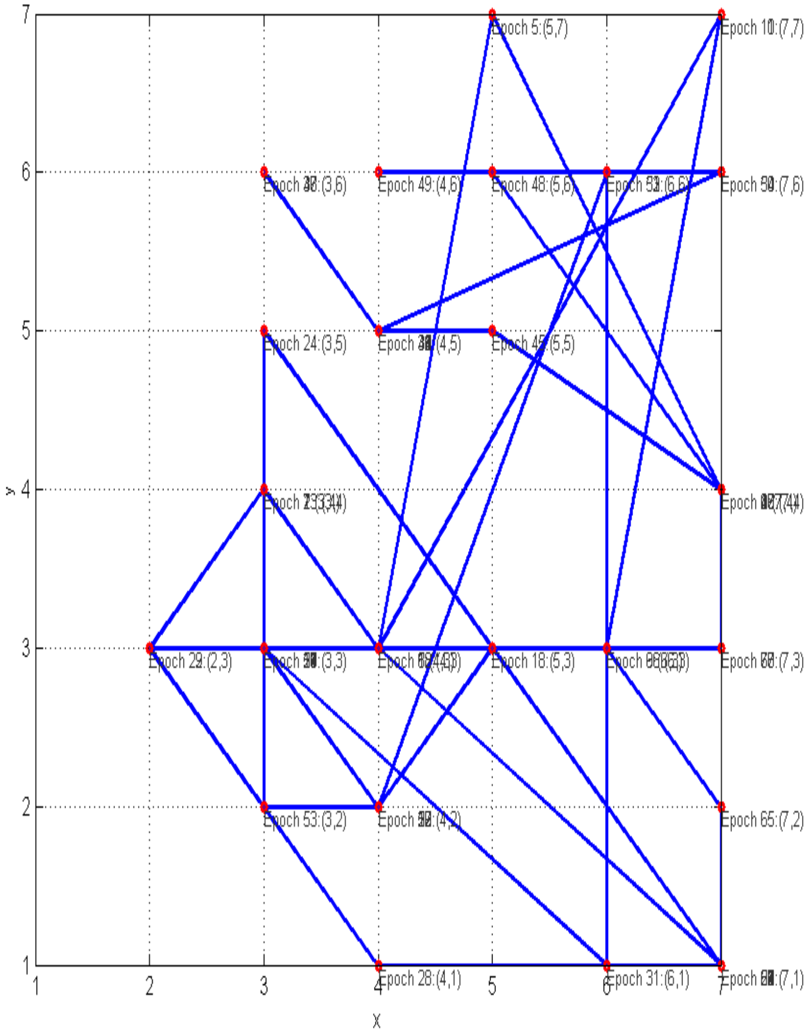


Figure 3.9: The Temporal Contextual Trajectory Descriptors of a Speech Utterance Using the MFCC Features Before Summarization.

To overcome these issues, we propose a novel method to summarize the trajectory, decrease redundancy, and reduce the contribution of noisy frames. The proposed method summarizes the trajectory by first decomposing the trajectory into T consecutive sub-trajectories. For each sub-trajectory, we compute its center of mass. In this case, the center of mass is defined as the most frequent position. Since the sub-trajectories inherent the redundancy, the most frequent position provides a stable measure for the center of mass and can be used to summarize the sub-trajectory and capture its dominant local behavior. A summarization of the trajectory is constructed by concatenating the center of mass of the consecutive sub-trajectories. The temporal ordering is still preserved in the new features since the sub trajectories are consecutive and ordered in time. Let $TCT = ((t_{i1}, t_{i2}), 1 \leq i \leq T)$ denotes the new temporal contextual descriptor.

The mapping of X using the Temporal Contextual Trajectory Algorithm is defined as follow

$$\Re^{N \times d} \rightarrow \Re^{T \times 2} \tag{16}$$

$$MAP(X) \rightarrow ((t_{i1}, t_{i2}), 1 \leq i \leq T) \tag{17}$$

In the rest of this thesis, we refer to the mapping of X using the Temporal Contextual Trajectory Algorithm by $TCT(X)$. Figure 3.10 represents the different steps of the approach. Figure 3.11 represents the example in figure 3.9 after summarization.

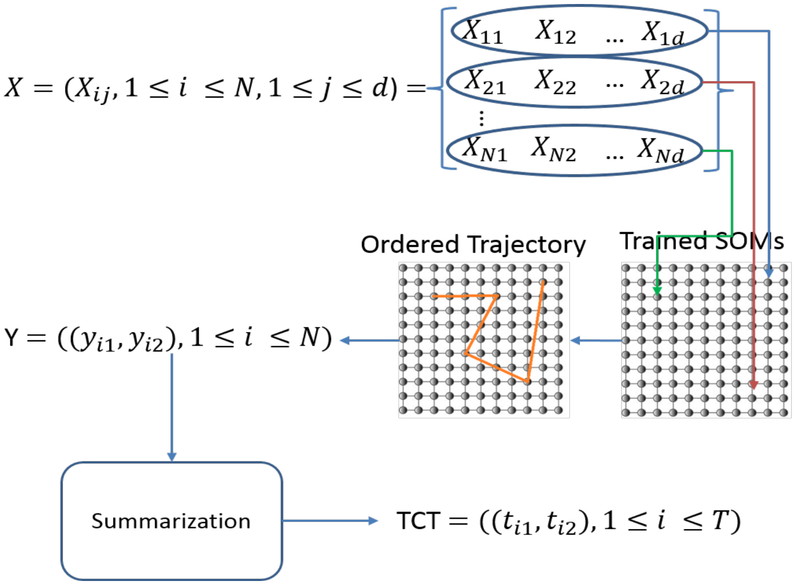

$X = (X_{ij}, 1 \leq i \leq N, 1 \leq j \leq d) =$
X_{11} X_{12} ... X_{1d}
X_{21} X_{22} ... X_{2d}
⋮
X_{N1} X_{N2} ... X_{Nd}
Ordered Trajectory
Trained SOMs
$Y = ((y_{i1}, y_{i2}), 1 \leq i \leq N)$
Summarization
$TCT = ((t_{i1}, t_{i2}), 1 \leq i \leq T)$

Figure 3.10: Architecture of The Temporal Contextual Trajectory Algorithm.

We used the same parameter settings and we set $T = 8$ resulting into a summarized trajectory with 8 epochs. As it can be seen, the summarized trajectory reproduces the behavior of the original trajectory while decreasing the length and the contribution of noise and redundancy. Figure 3.12 represents the comparison of the Temporal

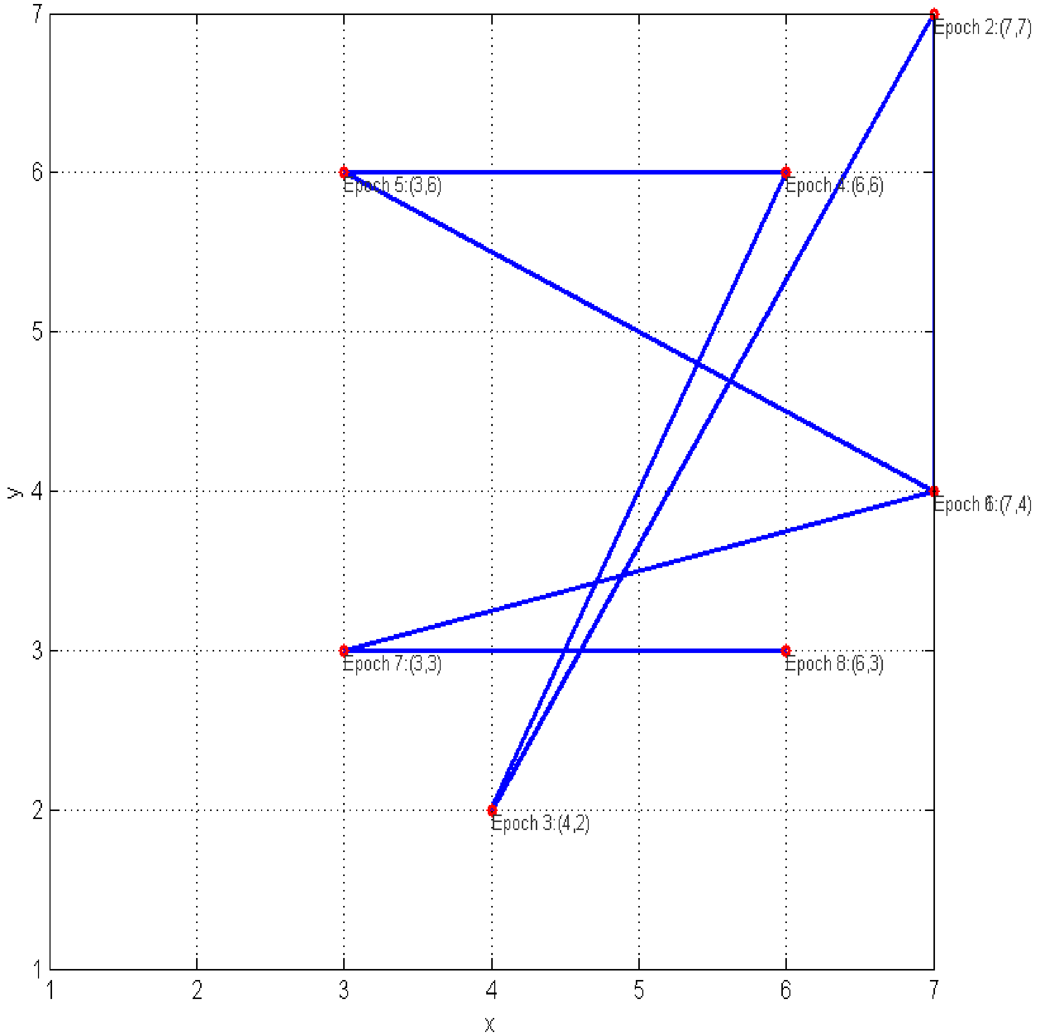


Figure 3.11: The Trajectory of a Speech Utterance mapped using the Temporal Contextual Trajectory Mapping Algorithm After Summarization.

Contextual Trajectory Descriptor of the same example in figure 3.6. We used the MFCC features with 13 coefficients. The frame size is set to 40 milliseconds with 50% overlap. The memory parameter is set to $w = 0.05$. We used a 7×7 map size for the SOMs and $T = 8$. Thus, the Temporal Contextual Trajectory Algorithm maps the sequence of MFCC features into one feature vector of dimension 8 (16 if we consider 2-D coordinates). As it can be seen, the Temporal Contextual Trajectory Descriptor of the two segments have different paths on the Map. In fact, the two

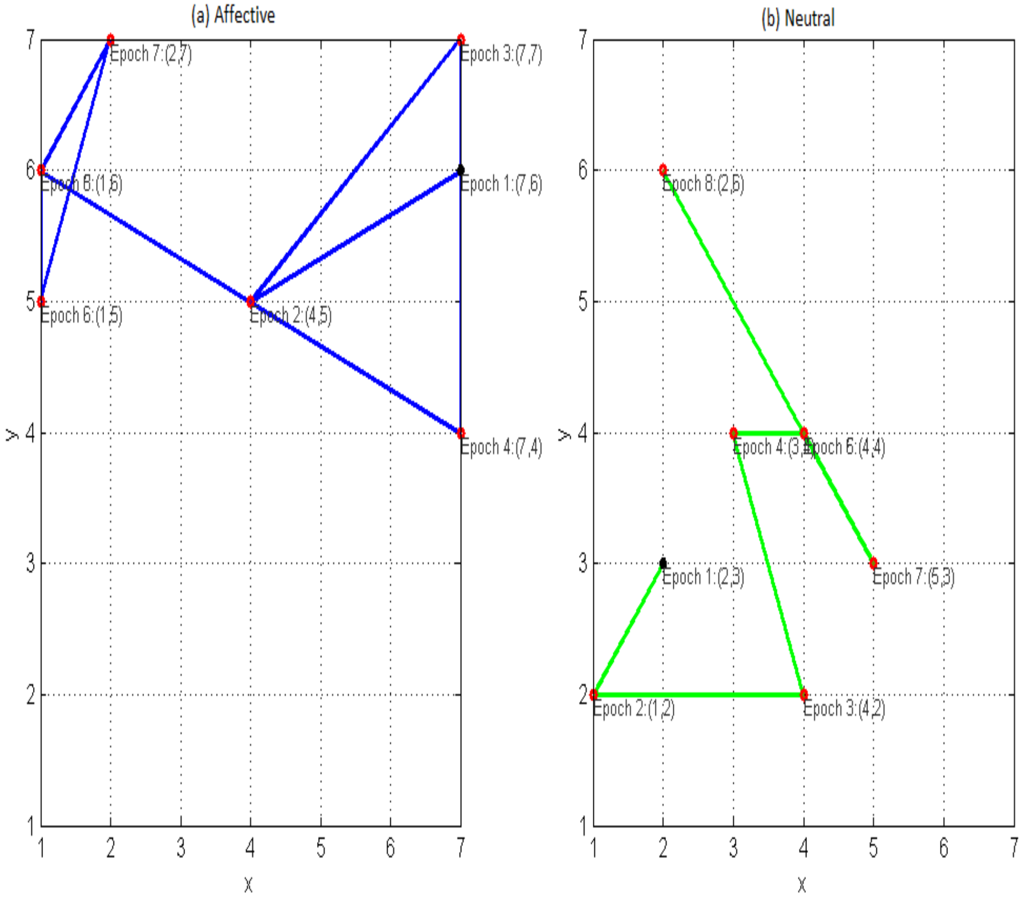


Figure 3.12: Comparison of the Temporal Contextual Trajectory Algorithm Descriptors of a Speech Utterance Expressed in Affective (a) and Neutral (b) States Using the MFCC Features.

trajectories occupy different regions of the map. In the case of the affective segment, the trajectory is mainly located in the upper area of the map. On the other hand, the neutral trajectory occupy the lower part of the map. Due to topology preserving of the SOMs, nearby map units are similar. Thus, map units labelled as affective are grouped together occupying specific regions of the map. Similarly, the neutral map units are located on different sides of the map. Typically, the class borders are located in the middle of the map. Thus, map units in the middle of the map are typically highly mixed and represent the similarity between the two classes. The difference in the trajectories represent the difference in the underlying emotion activity of the two utterances. Since the Temporal Contextual Descriptors are data-driven, we expect similar emotions to have similar trajectory behavior on the map while dissimilar emotions will have different trajectory behavior.

The Temporal Contextual Trajectory Descriptors preserves the temporal ordering of the original features by explicitly integrating it in the mapping.

The Temporal Contextual Trajectory Descriptors are also data driven and thus can be extended to learn from multiple emotional datasets for cross corpus and cross language emotion analysis.

The Temporal Contextual Trajectory Descriptors can also be used in real time emotion analysis. In that case, the input to the mapping algorithm is a sub-trajectory representing a texture window on which the analysis is applied on (in the order of 100 ms). Each sub-trajectory is processed and mapped on the SOMs. Then, the center of mass of the sub-trajectory is computed and appended it to a global trajectory. The global trajectory represents the Temporal Contextual Trajectory of the entire real time speech stream. Thus, it can be used to monitor and detect emotions in real time settings.

In order to improve the representation of emotions on the map, first, we can label the map units of the SOMs using the training data to reflect a certain emotion. The resulting map can be thought of as a 2-D map of emotions. Second, techniques such as Learning Vector Quantization[70] can be used to rearrange and reorder the map units of the SOMs to maximize the discrimination between emotions.

3.3 Overview of the Proposed Speech Emotion Recognition and Detection System

The proposed framework maps temporal data into a new descriptor as described in the previous section. Our goal is to show that the new descriptors are efficient for representation, analysis, and classification of emotions. In order to achieve this goal, we incorporated the proposed framework into a speech emotion detection and recognition system. The goal of such system is to index emotional databases to facilitate the classification and retrieval of emotions. An illustrative block diagram of the proposed system is shown in figure 3.13.

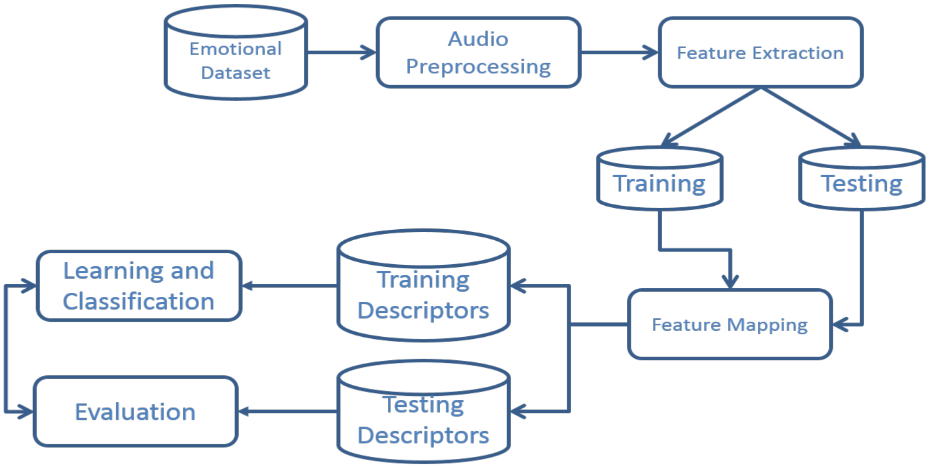


Figure 3.13: Architecture of The Proposed Speech Emotion Detection and Recognition System.

3.3.1 Preprocessing

Typically, the audio signals of utterances are usually noisy due the background and hiss of the recording machine. Noise corrupts the signal and consequently deteriorates the performance of the subsequent steps.

3.3.1.1 Power Substraction Filtering

The emotional corpus is preprocessed in order to reduce the contribution of background noise. We used the power substraction algorithm described in Appendix A to filter the audio signals. Figure 3.14 represents a speech utterance and its filtered version using the power substraction algorithm. We used the first 100ms of the signal to estimate the noise power spectrum which represent mostly background and recording conditions noise. Then, we subtracted the noise from the input signal as described in appendix A.

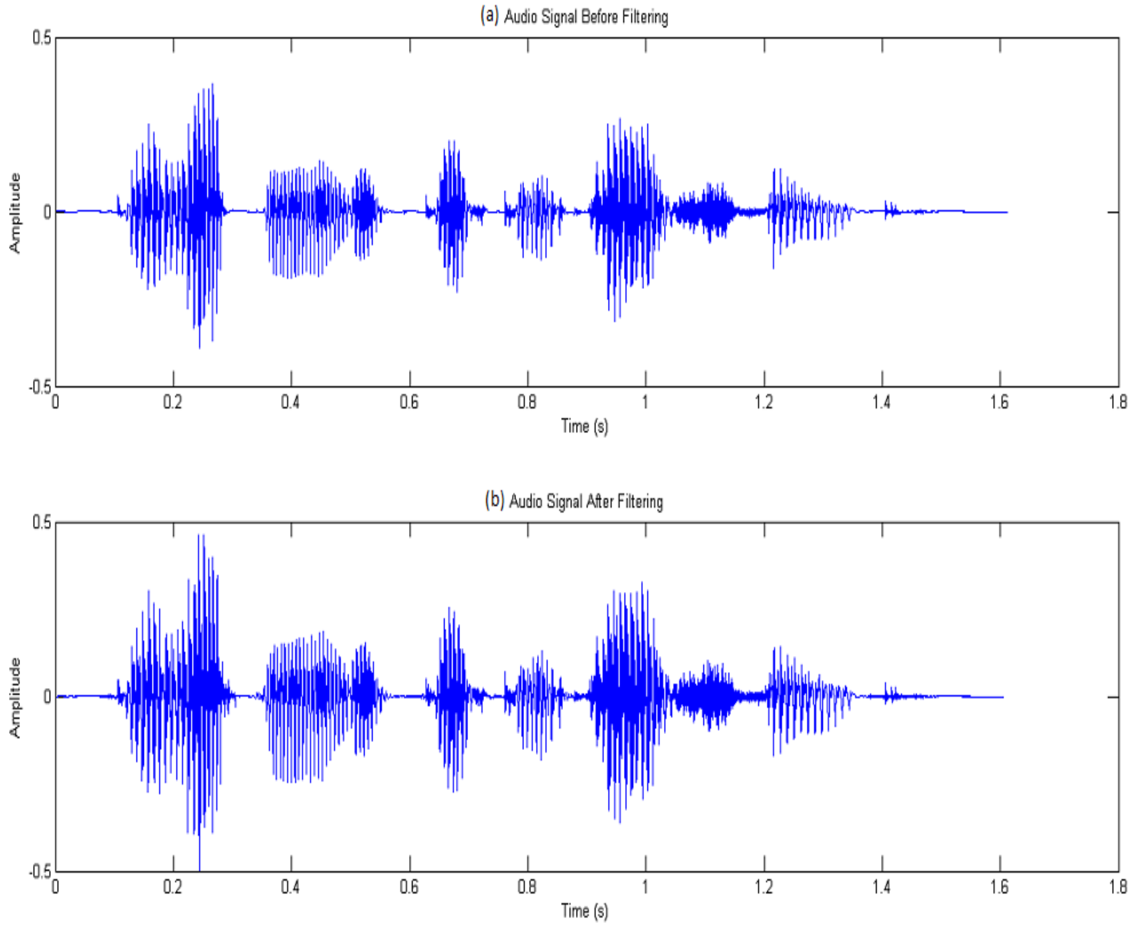


Figure 3.14: Comparison between an audio signal (a) and its filtered version computed using the Power Substraction Algorithm (b).

3.3.2 Feature Extraction

After audio preprocessing, acoustic features are extracted from each utterance in the corpus. Recalling the notation in section 2.3, each utterance in the corpus is divided into overlapping frames. For each frame, we extract F acoustic features. Let $\mathbf{x}$ denotes the input utterance. Let $X^{(f)} = (X_{ij}^{(f)}, 1 \leq i \leq N, 1 \leq j \leq d_f)$ denotes the feature representation of $\mathbf{x}$ using acoustic feature f of dimension d_f where $1 \leq f \leq F$. The acoustic features can be of any configuration or dimension as long as they are temporal.

3.3.3 Mapping

After feature extraction, the corpus is mapped to temporal contextual descriptors using the proposed framework introduced in the previous section. For each acoustic feature f, the feature representation of each audio segment is mapped into three new descriptors using the three proposed mapping algorithms. We refer to the global descriptor computed by the Temporal Averaging Algorithm for input utterance $\mathbf{x}$ using acoustic feature f by $\mathrm{TAA}(X^{(f)})$. We refer to the global descriptor computed by the Temporal Response Averaging Algorithm for input utterance $\mathbf{x}$ using acoustic feature f by $\mathrm{TRAA}(X^{(f)})$. We refer to the global descriptor computed by the Temporal Contextual Trajectory Algorithm for input utterance $\mathbf{x}$ using acoustic feature f by $\mathrm{TCT}(X^{(f)})$. For simplicity, we refer to $\mathrm{MAP}(X^{(f)})$ by the mapping of $X^{(f)}$ by any of the three algorithm mentioned in this paragraph. That is MAP refer to TEMP, RIM, or TCT. Figure 3.15 represents the architecture of the feature extraction and mapping for an input audio signal $\mathbf{x}$.

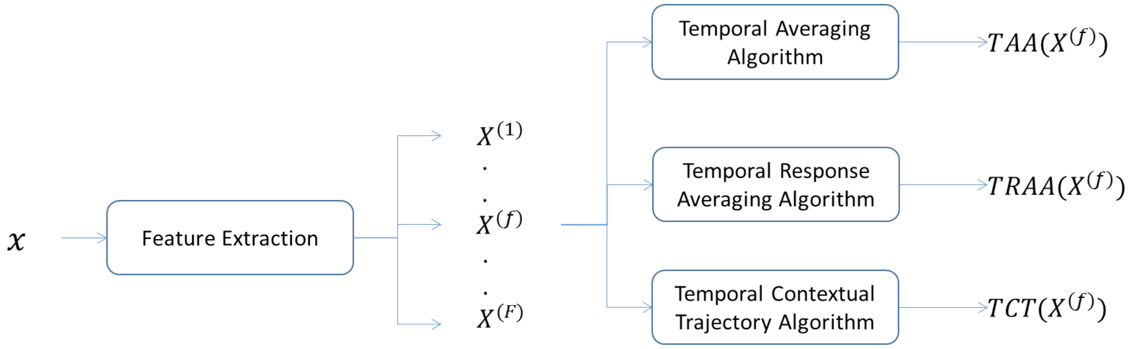

x
Feature Extraction
$X^{(1)}$
$X^{(f)}$
$X^{(F)}$
Temporal Averaging Algorithm
Temporal Response Averaging Algorithm
Temporal Contextual Trajectory Algorithm
$TAA(X^{(f)})$
$TRAA(X^{(f)})$
$TCT(X^{(f)})$

Figure 3.15: Architecture of Feature Extraction and Mapping.

3.3.4 Classification and Learning

After mapping, the descriptors are presented to a classifier for learning and classification. In this thesis, we use a standard feed-forward network with two hidden layers [71]. The mapped features are split into training, testing, and validation sets. The training set is used by the system for learning the parameters of mapping and learning. We used the scaled gradient back-propagation algorithm [72] to train the classifier. The validation set is used to measure network generalization, and to halt training when generalization stops improving. The testing set is used to measure the performances of the system after training. A different classifier is used for each mapping algorithm and each acoustic feature.

3.3.5 Fusion

As discussed in the previous paragraph, a different classifier is used for each mapping algorithm and each acoustic feature. After classification, for each mapping algorithm, the decisions of the different classifiers are combined to compute a global decision about the underlying emotion of the input audio segment.

In this thesis, we derive, use, and compare two score level fusion methods. Our choice of using score level fusion is motivated by two reasons.

First, Neural Networks output a score vector for each input sample. Thus, it is more intuitive to apply score level fusion.

Second, score level fusion is widely used in emotion analysis. Thus, it provides a good baseline to compare the performance of our system against existing methods in the literature. In the next subsections, we introduce the two score level fusion methods.

Recalling that F denotes the number of acoustic feature sets used in the system, each mapping algorithm is represented by F classifiers. The output of each classifier is a $1 \times M$ score vector where M denotes the number of emotions or classes.

Let $\mathbf{x}$ denotes an input utterance. Let $X^{(f)} = (X_{ij}^{(f)}, 1 \leq i \leq N, 1 \leq j \leq d_f)$ denotes the feature representation of $\mathbf{x}$ using acoustic feature f of dimension d_f where $1 \leq f \leq F$. Let $\{MAP(X^{(f)}), 1 \leq f \leq F\}$ represents the set of F feature vectors of utterance x after feature mapping. MAP denotes one of the mapping algorithms proposed in this thesis. That is, MAP refers to TAA, TRAA, or TCT. For simplicity, we replace MAP($X^{(f)}$) by $\mathbf{y}_f$. That is, $\{MAP(X^{(f)}), 1 \leq f \leq F\}$ is now referred to by $\{\mathbf{y}_f, 1 \leq f \leq F\}$.

Let $S_j(y_f)$ denotes the score assigned by classifier f to the feature representation y_f of $\mathbf{x}$ for class j where $\sum_{j=1}^{M} S_j(\mathbf{y}_f) = 1$. Since the score vector computed by each classifier sums to one, the score assigned by each classifier to an input vector can be considered as a posterior probability. That is, $S_j(\mathbf{y}_f) = P(C_j/\mathbf{y}_f)$ where C_j denotes class or emotion j. The combination of the F classifiers must maximize the overall posteriori probability $P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)$ where $\{1 \leq j \leq M\}$.

3.3.5.1 Summation Rule Score Fusion

In the first approach, we formulate the fusion as a summation rule score fusion. The most used approach computes a global score for each class. In this case, the average of the score vectors of the F classifiers is computed and the input feature vector is assigned to the class with the highest average. That is

$$Assign\, \mathbf{x} \;\; \rightarrow \; C_k \; if \tag{18}$$

$$P(C_k/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \max_{j=1}^{M} \frac{1}{F} \sum_{i=1}^{F} S_{ij}(\mathbf{x}) \tag{19}$$

Other statistics such as max, median, and min can also be used to compute the global score for each class.

3.3.5.2 Product Rule Score Fusion

In the second approach, we formulate the score level fusion as a product rule. The input audio segment x is assigned to the class with the highest posteriori probability. That is

$$Assign\,\mathbf{x} \quad \rightarrow \; C_k\; if \tag{20}$$

$$P(C_k/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \max_{j=1}^{M} P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) \tag{21}$$

Using Bayes theorem

$$P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \frac{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F/C_j)P(C_j)}{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)} \tag{22}$$

The F classifiers are independent. Thus,

$$P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F/C_j) = \prod_{f=1}^{F} P(\mathbf{y}_f/C_j) \tag{23}$$

We replace $P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F/C_j)$ in equation 22 by its formula in 23

$$P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \frac{P(C_j)\prod_{f=1}^{F} P(\mathbf{y}_f/C_j)}{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)} \tag{24}$$

Applying Bayes Theorem again, we obtain

$$P(\mathbf{y}_f/C_j) = \frac{P(C_j/\mathbf{y}_f)P(\mathbf{y}_f)}{P(C_j)} \tag{25}$$

We replace $P(\mathbf{y}_i/C_j)$ in equation (24) by its formula in (25). We get

$$P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \frac{P(C_j)\prod_{f=1}^{F} \frac{P(C_j/\mathbf{y}_f)P(\mathbf{y}_f)}{P(C_j)}}{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)} \tag{26}$$

That is

$$P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \frac{P^{-(F-1)}(C_j)\prod_{f=1}^{F} P(C_j/\mathbf{y}_f)P(\mathbf{y}_f)}{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)} \tag{27}$$

Equation (27) represents the computational form of the overall posteriori probability in function of the posteriori probability of the individual F classifiers. For each input utterance x, we assign it to the class with the highest overall posteriori probability. That is

$$
\begin{aligned}
&Assign\, \mathbf{x} \quad \rightarrow \; C_k\; if && (28)\\
&P(C_k/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \max_{j=1}^{M} P(C_j/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) && (29)\\
&\qquad = \max_{j=1}^{M} \frac{P^{-(F-1)}(C_j) \prod_{i=1}^{F} P(C_j/\mathbf{y}_i) P(\mathbf{y}_i)}{P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)} && (30)
\end{aligned}
$$

Since $P(\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F)$ is independent of the classifiers used, we focus only on the numerator of equation (30). That is

$$
\begin{aligned}
&Assign\, \mathbf{x} \quad \rightarrow \; C_k\; if && (31)\\
&P(C_k/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \max_{j=1}^{M} P^{-(F-1)}(C_j) \prod_{f=1}^{F} P(C_j/\mathbf{y}_f) P(\mathbf{y}_f) && (32)
\end{aligned}
$$

In terms of score, equation (32) can be rewritten as

$$
\begin{aligned}
&Assign\, \mathbf{x} \quad \rightarrow \; C_k\; if && (33)\\
&P(C_k/\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_F) = \max_{j=1}^{M} P^{-(F-1)}(C_j) \prod_{f=1}^{F} S_{fj}(x) P(\mathbf{y}_f) && (34)
\end{aligned}
$$

As it can be seen in equation (34), the overall score is the product of the individual posteriori probability. The fusion is a "severe" score level fusion. That is, the function form in equation (34) is highly sensitive to the individual scores of the classifiers. In fact, if one of the classifier assign a low score to the input vector then the overall product will also be low or close to zero. In the next two chapters, we evaluate and compare the performances of the two fusion techniques.

CHAPTER 4

EXPERIMENTAL RESULTS: BERLIN DATASET

In the previous chapter, we introduced our feature mapping framework. We also integrated the framework into a speech emotion detection and recognition system. In this chapter, we evaluate the performances of the proposed work using an acted dataset, namely, the Berlin Emotional Dataset (BED). As previously mentioned, in this scenario, the emotional speech is acted by subjects in a professional manner. Such corpus provides a good baseline to evaluate the performances of our feature mapping framework against the Statistics Based Mapping Algorithm. The chapter is organized as follow:

First, we introduce the data collection and the parameters settings used for the experiments. Second, we compare and analyze the performances of the proposed framework on the BED against the Statistics Based Mapping Algorithm. Third, we analyze and report the classification performances of the proposed emotion detection and recognition system.

4.1 Corpus and Parameters Settings

4.1.1 Corpus

In this chapter, we use the Berlin Emotional Dataset to evaluate the performances of our system. The corpus is widely used in emotion analysis [3, 11, 12, 5, 35]. The database consists of ten utterances produced by 5 males and 5 females (10 subjects) in German. The transcripts of the different utterances in German and English are displayed in table 4.2. We refer to each utterance in the corpus by its code throughout this chapter (For instance, the first utterance is referred by $a01$). The details of the subjects (speakers) is provided in table 4.3. We refer to each subject in the corpus by his/her code throughout this chapter (For instance, the first speaker is referred by 03). The subjects acted 7 different emotions namely (hot) anger, happiness, fear (panic), sadness (sorrow), boredom, disgust, and neutral. The sentences were taken from everyday communication and could be interpreted in all emotional contexts without semantic inconsistency. The number of utterances for each class is displayed in table 4.4.

TABLE 4.2

The Transcript of The Utterances in The Berlin Emotional Dataset.

Code	Text
a01	Der Lappen liegt auf dem Eisschrank.
	The tablecloth is lying on the fridge.
a02	Das will sie am Mittwoch abgeben.
	She will hand it in on Wednesday.
a04	Heute abend knnte ich es ihm sagen.
	Tonight I could tell him.
a05	Das schwarze Stck Papier befindet sich da oben neben dem Holzstck.
	The black sheet of paper is located up there besides the piece of timber.
a07	In sieben Stunden wird es soweit sein.
	In seven hours it will be.
b01	Was sind denn das fr Tten, die da unter dem Tisch stehen?
	What about the bags standing there under the table?
b02	Sie haben es gerade hochgetragen und jetzt gehen sie wieder runter.
	They just carried it upstairs and now they are going down again.
b03	An den Wochenenden bin ich jetzt immer nach Hause gefahren und habe Agnes besucht.
	Currently at the weekends I always went home and saw Agnes.
b09	Ich will das eben wegbringen und dann mit Karl was trinken gehen.
	I will just discard this and then go for a drink with Karl.
b10	Die wird auf dem Platz sein, wo wir sie immer hinlegen.
	It will be in the place where we always store it.

The recordings were carried out in separate sessions in an anechoic chamber using a Sennheiser MKH 40 P 48 microphone and a Tascam DA P1 portable DAT-recorder. The emotional content of the speech material and its naturalness were evaluated by

TABLE 4.3

Details of The Subjects in The Berlin Emotional Dataset.

Code	Gender	Age
03	male	31 years
08	female	34 years
09	female	21 years
10	male	32 years
11	male	26 years
12	male	30 years
13	female	32 years
14	female	35 years
15	male	25 years
16	female	31 years

TABLE 4.4

The Number of Utterances per Emotion in The Berlin Emotional Dataset.

	Anger	Boredom	Disgust	Anxiety/Fear	Happiness	Sadness	Neutral
Number of Samples	127	81	46	69	71	62	79

20 naive listeners and each utterance is given a score representing its authenticity to reproduce the emotion. The given score can be used as a ground truth in the analysis. Table 4.5 illustrates the details of the Berlin emotional collection.

Our motivation to use the Berlin dataset is due to its public availability and wide use in speech emotion analysis [3, 11, 12, 5, 35].

TABLE 4.5

The Details of the Berlin Emotional Dataset.

Number of Classes	Number Of Subjects	Sampling Rate (kHZ)	Corpus Type
7	10 (5 male and 5 female)	22050	Acted

4.1.2 Taxonomy

In this thesis, we propose 3 different taxonomies for the Berlin emotional dataset. Figure 4.16 represents the proposed taxonomies of the Berlin emotional dataset. The three different taxonomies are the following:

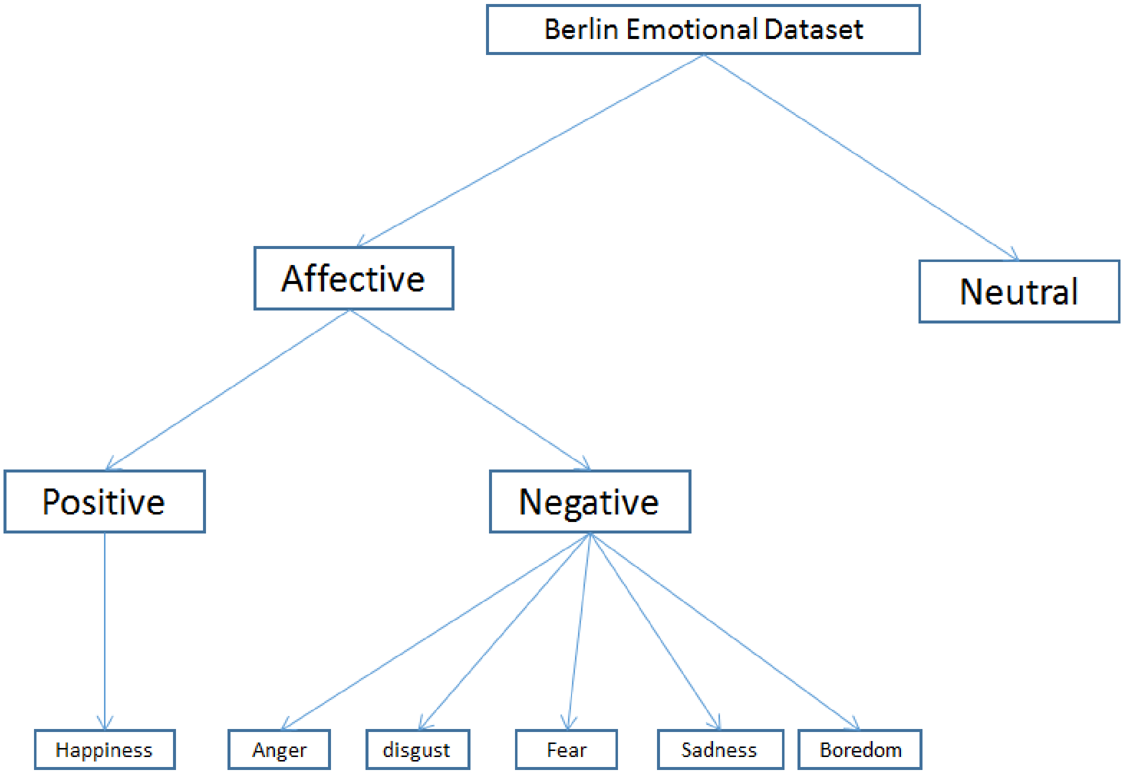


Figure 4.16: The Different Taxonomies of The Berlin Emotional Dataset

Neutral Vs. Affective Emotions: The corpus can be organized into two different classes namely: Neutral and Affective states. The affective class contains all the classes except neutral which is represented by its own class. The goal of this scenario is the detection of emotions in speech. This scenario can be used to detect confident speakers in debates, detect truth/lie, etc.

Positive Vs. Negative Emotions: We organized the corpus into three different classes namely: positive, negative, and neutral. The positive class contains happiness. The negative class contains anger, fear, sadness, boredom, and disgust.

This scenario can be used in call centers to detect and recognize negative emotions from callers/customers. The negative class contains 5 subclasses. The positive class contain 1 subclass.

7 Emotions: We also used the corpus in its default taxonomy. The goal of this scenario is the recognition of emotions from speech.

4.1.3 Parameters Settings

There are different parameters that govern the conducted experiments. In this subsection, we discuss these different parameters and their influence on the proposed system and experiments.

4.1.3.1 Frame and Hop Size

The acoustic features used in our system are extracted at the frame level. Another parameter that needs to be chosen a priori is the length of the frame and its hop-size. These parameters affect the time and the frequency resolution of the analysis and there is no universal rule to choose them. Typically, the frame and hop size are in the order of milliseconds (10 ms -100 ms) [20, 15]. We have conducted several experiments with different frame and hop size. We found out that the results did not vary vastly when a window size between 20 ms and 60 ms. Thus, the choice of the frame and hop size is mainly based on computational efficiency. In our case, we concluded that a window size of 40 ms with 50% overlap (that is 20 ms as the hop size) provides a good balance between computational complexity and efficiency. The same window and hop size are used throughout this chapter.

4.1.3.2 Training and Learning

The Berlin Emotional Corpus is divided into training, testing, and validation sets. The training set is used to estimate the parameters of the mapping and the learning model. The testing set is used to evaluate the performances of the system. The validation set is used by the neural network to validate the learning model of the classifier. In all experiments, 70%, 15%, and 15% are used for training, testing, and validation respectively.

4.2 Feature Extraction

Various features have been proposed in the literature of speech emotion detection and recognition. The choice of the feature sets in the system must be based on discrimination between emotions.

Our feature mapping framework works on any short term-based acoustic features. In this thesis, we use common acoustic features to demonstrate the performances of our framework. In our experiments, we used 5 feature extraction algorithms namely: MFCC, ΔMFCC, Pitch (F0), Perceptual linear predictive (PLP) Rasta , and Low Level Spectral and Temporal Features. In the next few subsections, we discuss these 5 acoustic features used in the experiments.

4.2.1 MFCC

The performance of the MFCC feature extraction algorithm depends on various parameters such as the number, the shape, the spacing of the filters used in the mel filter bank analysis, and the warping of the power spectrum. In [73], the authors conducted a comparative study of 4 different implementations of the MFCC features. The 4 algorithms differ in the approximation of the nonlinear pitch perception, the filter bank design, and the compression of the filter bank output. They used text-independent speaker verification to compare the performance of the different implementations in the literature. They found out that the performance of the 4 algorithms did not vary vastly when different approximations of the parameters discussed above are used. They concluded that regardless of the filter design, a large number of filters increases the performance of speaker detection. They also found out

that the spacing between the filters proved to be a sensitive parameter (increasing or decreasing the overlap of the filters beyond a given range increases the error rate). In our system, we use the MFCC FB-40 feature extractor explained in [74] with the same parameters in [73].
Various methods proposed in the literature use the MFCC for emotion analysis [9, 20, 15]. In [75], the authors conducted a study about the use of MFCC for emotion recognition. The authors concluded that statistics computed from MFCCs carry emotional information. Our choice of the MFCC features is motivated by its wide use in the literature. We conducted various experiments to investigate the optimal number of MFC coefficients. We concluded that 13 provides a good balance between computational complexity and performance. Thus, the number of MFC coefficients is set to 13.

4.2.2 ΔMFCC

Typically, the MFCC features are further processed to compute another acoustic feature named ΔMFCC. The ΔMFCC is defined as the first derivative of the MFCC. The Delta coefficients are computed via a linear regression formula defined as:

$$\Delta c[m] = \frac{\sum_{i=1}^{N} i.(c[m+i] - c[m-i])}{2\sum_{i=1}^{N} i^2} \tag{35}$$

In (35), N is the size of the regression window. $c[j]$ denotes the j^{th} MFCC coefficient. In our experiments, we tested various values of the regression window size N ranging between 2 and 9 (typical values for N). We found out that increasing N actually decreases their discrimination power. In fact, by increasing N, the number of MFCC coefficients involved in the computation in equation 35 increases and the ΔMFC coefficients tend to be more uniform. We conclude that $N = 2$ provides the best discrimination power and computational efficiency.

4.2.3 Pitch

Pitch detection is an important task in many audio analysis such as speaker recognition, emotion analysis, speech analysis synthesis (vocoder), and music analysis. Because of its importance, various algorithms for pitch detection have been proposed in the literature [25, 26, 27]. These methods can be categorized into three groups namely: time domain, frequency domain, and hybrid (combination of both). The first category uses the time domain properties of the speech signal to estimate the pitch. The second uses the frequency domain properties of the speech signal. The third group is a combination of the time and frequency domain. Many studies were conducted to compare the performances of the pitch detection algorithms proposed in the literature [25, 26, 27]. In [25], the authors extensively reviewed 7 different algorithms for the extraction of pitch information. They reviewed 4 time-domain, one frequency domain, and two hybrid pitch detectors. They used a series of experiments to compare the performances of the 7 pitch detectors based on various performance criteria such as accuracy in estimating the pitch period, robustness, computational complexity. For that purpose, they proposed 5 different error measures to estimate the performances of each algorithm. They concluded that no single pitch detector was uniformly ranked the top across all experiments. Instead, each algorithm has advantages and weaknesses. They concluded that time domain and hybrid pitch detectors are not efficient for detecting low pitch values due to the use of short analysis frame size. They concluded also that spectral pitch detector are not efficient for detecting high pitch values due to the small number of harmonics which are present in their spectra.

In [26], the authors conducted another study on 6 modern pitch detector algorithms. They used a large dataset of singing sounds to investigate the performances of each algorithm. They proposed 4 different error metrics introduced in [76] for the performances criteria. They concluded that there is no single pitch detector that outperform the other ones and that each algorithm has advantages and drawbacks depending on the error metric used. Comparing the different pitch detector algorithms proposed in the literature is out of the scope of this thesis. We use the auto-correlation method [26] to estimate the pitch information.

4.2.4 Low Level Features

The low level features used in our experiments are energy entropy, short time energy, zero cross rate, spectral roll off , spectral flux, and spectral centroid [29]. The low level features represent various statistics computed from the audio signal and its spectrum representation. The details of the feature extraction algorithms of the six low level features are provided in appendix A. The dimension of low level features is 6.

4.2.5 PLP-Rasta

The Relative Spectral Transform (RASTA)- Perceptual Linear Prediction (PLP) is a widely used speech feature representation [77]. Originally, the PLP was introduced in [77]. Perceptual linear prediction, similar to linear predictive analysis discussed in section 2.5.1 , is based on the short-term spectrum of speech. However, perceptual linear prediction(PLP) modifies the short-term spectrum of the speech by several psychophysically based transformations. The PLP features are sensitive to spectral distortions. These are due to the noise added by the communication channel. On other hand, RASTA is a smoothing technique that applies a band-pass filter to the energy in each frequency sub-band in order to smooth over short-term noise variations and to remove any constant offset resulting from static spectral coloration in the speech channel e.g. from a telephone line [78]. The combination of RASTA and

PLP provides a feature representation robust to linear spectral distortions caused by communication channels. A detailed implementation of the RASTA-PLP features can be found in [79]. In our experiments, we used 12 as the order of prediction. Thus, the dimension of RASTA-PLP features is 13.

4.3 Feature Mapping

In this section, we compare the performances of the proposed framework against the Statistics Based Mapping Algorithm (SBMA).

Many statistics measures have been investigated and used in the Statistics Based Mapping Algorithm. The most common are max, min, mean, standard deviation, and median [20, 15]. In this thesis, we use max, min, mean, standard deviation, median, and max divided by min. We refer to such descriptors throughout this chapter as SBMA Descriptors. Recalling the notation in the previous chapter, we refer to the global descriptor computed by the Temporal Averaging Algorithm by TAA Descriptors. We refer to the global descriptor computed by the Temporal Response Averaging Algorithm by TRAA descriptors. We refer to the global descriptor computed by the Temporal Contextual Trajectory Algorithm by TCT descriptors.

For simplicity, the experiments conducted in this section are conducted on the MFCC features only. The same experiments can be conducted using the other acoustic features. Figures 4.17 and 4.18 represent an utterance $a01$ spoken by subject 03 in neutral and affective state respectively and their TAA, TRAA, TCT, and SBMA descriptors. The memory parameter w is set to 0.05 for the TAA and TRAA descriptors. The SOMs map size is set to 7×7 for the TRAA and TCT descriptors. The response for the TRAA descriptors is computed in a crisp fashion. For each utterance, we compute the TAA (1×13), TRAA (1×49), TCT (8×2) (In this example, we use the 2-D coordinate for display), and the SBMA (1×78) descriptors.

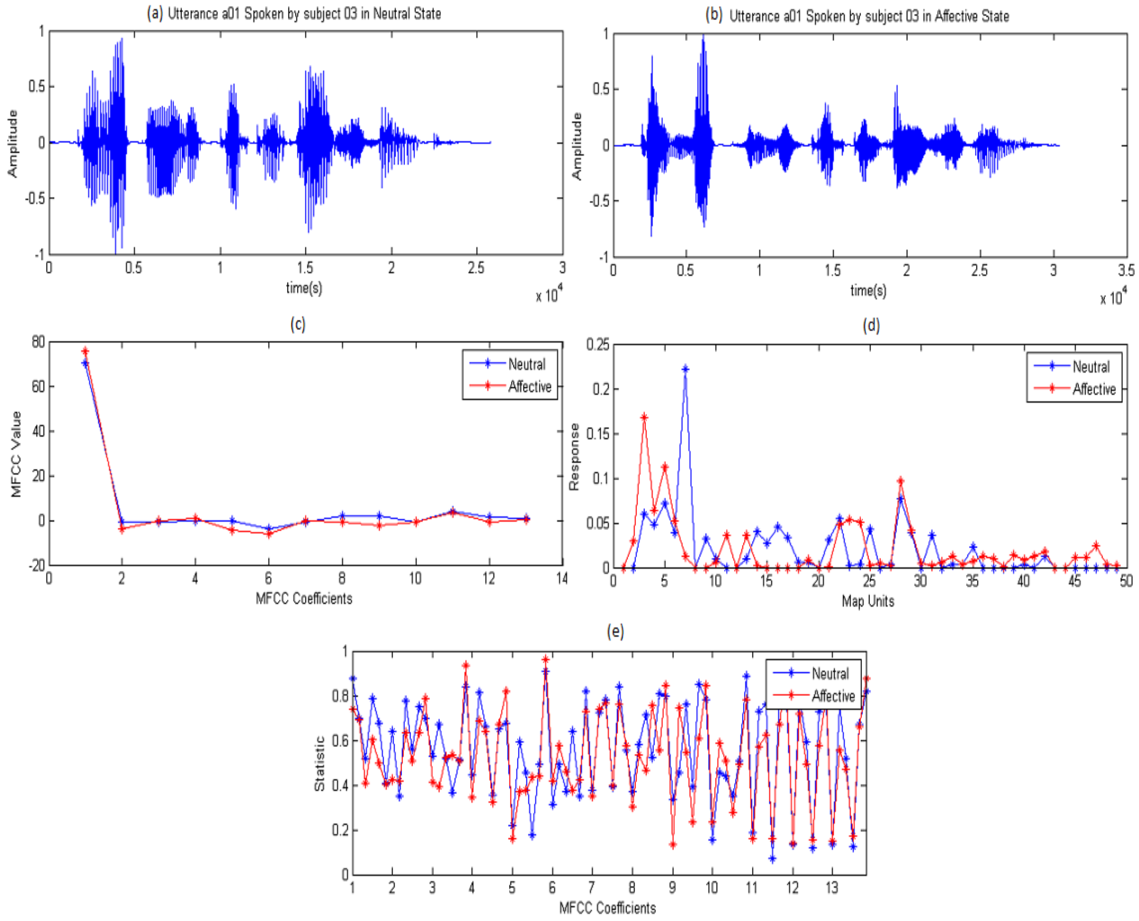


Figure 4.17: Comparison of the TAA (c), TRAA (d), and SBMA (e) descriptors using the MFCC features for the utterance $a01$ spoken by subject 03 in neutral (a) and affective (happiness) (b) states.

The different parameters of the experiment are summarized in table 4.6. The same settings are used throughout this section unless we mention it. The two utterances represent the same sentence $a01$ and spoken in neutral and affective (Happiness) by the same subject 03. Thus, we expect strong similarity in their representation except where emotion is produced. As it can be seen, the TAA descriptors for both utterances are strongly correlated in some coefficients and dissimilar in other coefficients such as coefficient 1, 5, and 9. This dissimilarity suggests that the emotional activity is captured in those specific coefficients. The TRAA descriptors also have different distributions on the map units. In fact, the distributions are correlated expect for

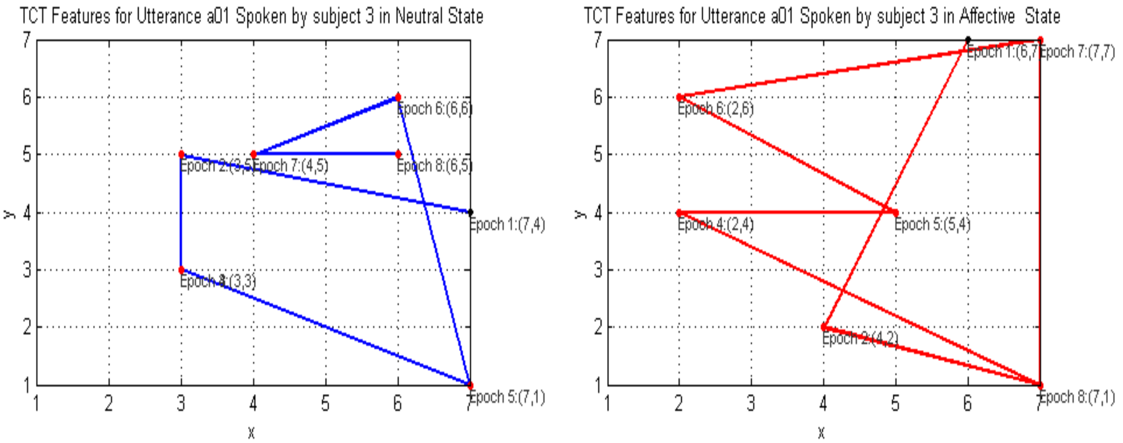


Figure 4.18: Comparison of the TCT descriptors using the MFCC features for the utterance $a01$ spoken by subject 03 in neutral and affective (happiness) states.

TABLE 4.6

Parameter Settings for the Experiments

Parameter	Value
Acoustic Feature	MFCC
Number of MFC coefficients	13
SOMs Map Size	7 × 7
Number of Classes	2
Frame Size	40 ms.
Hop Size	20 ms.
Memory w	0.05
Response Computation	Crisp
Dimension of RIM Features	1 × 49
Dimension of TEMP Features	1 × 13
Dimension of TCT Features	8 × 2
Dimension of STD Features	1 × 78

few map units such as map unit 7. The map units of the SOMs are clusters that are labeled and represent specific emotions. High responses to affective clusters suggest that there is emotional activity. In the TCT descriptors, the trajectory of both utterances share common positions. The TCT descriptors are ordered in time. Thus, we can notice that the two trajectories started around the same location. The beginning of the trajectories are usually silence segments before speaking the utterance. Moreover, the affective trajectory is more variable on the map than the neutral utterance.

The two trajectories ended in different areas of the map.

In the SBMA descriptors, the statistics are highly correlated for all the MFCC coefficients except for coefficient 1, 5, and 9 (same as the TAA descriptors). However, since the descriptors are static in time, they don't convey any information about the acoustic activity leading to the emotion or the time when the affective utterance is different from the neutral version. Moreover the SBMA descriptors are ambiguous for display.

The true label of the affective utterance used in the previous example is happy. The same conclusions can be concluded for any other affective state. Figures 4.19 and 4.20 represent the same example used in the previous paragraph but this time we replaced the affective utterance with the angry version.

As it can be seen, the same conclusions can be derived about the three proposed descriptors. The TRAA and TCT descriptors outperform the SBMA descriptors in discriminating between the two utterances. The TCT descriptors are similar in the beginning and end of the trajectory for the two utterances. That is due to the easy noticeable voice inactivity at the end of the audio signal of the two utterances. The affective utterance is highly variable compared to the neutral version. That is due to the emotional arousal due to anger. In fact the MFC coefficients represent the energy computed on the Mel filter. Emotions such as anger are characterized by sudden changes in tone, voice which are reflected by changes in energy which explains the variability. The TRAA descriptors have different distributions for both utterances. Each utterance has a high response to map units where the other utterance have insignificant response. Similarly, the different between the affective and neutral TAA descriptors is noticeable. Moreover, the three proposed descriptors are more suitable for display and visualization which improve the analysis of emotions.

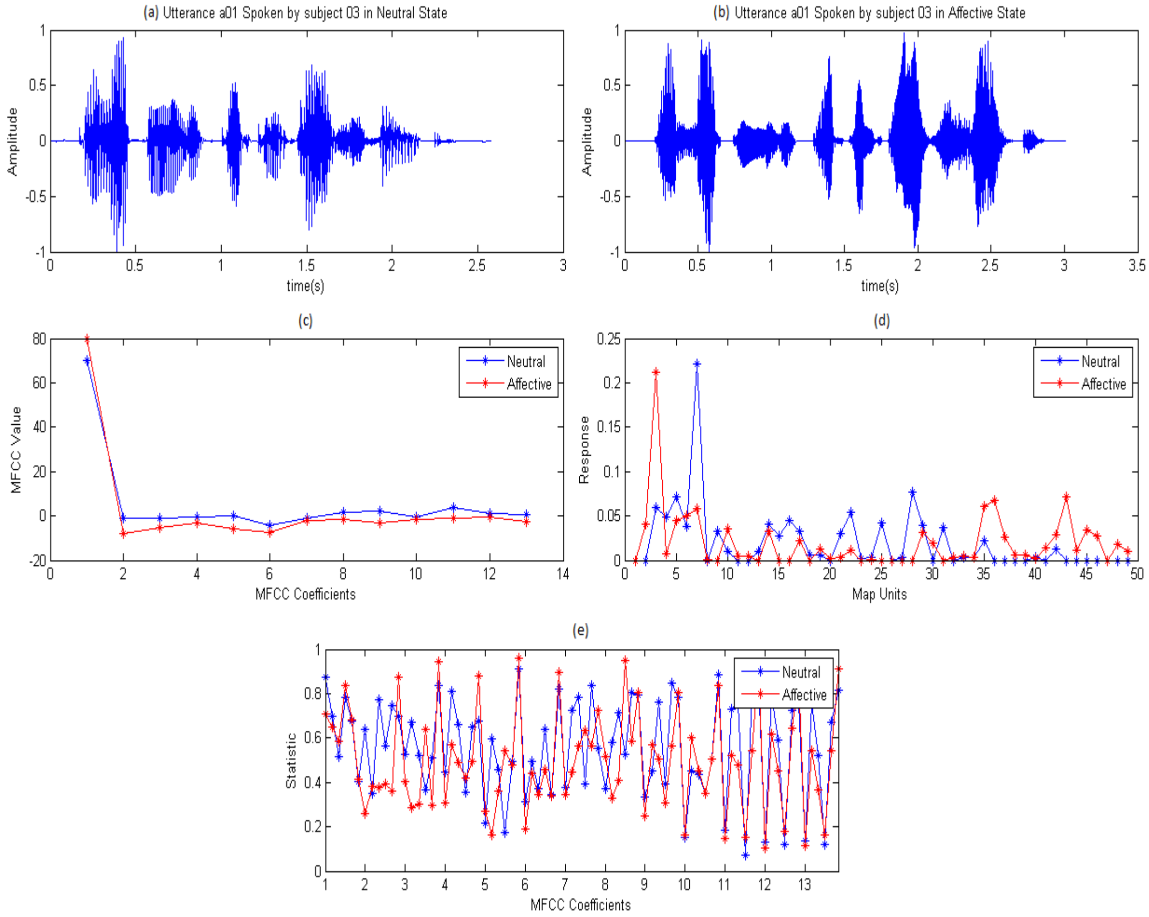


Figure 4.19: Comparison of the TAA (c), TRAA (d), and SBMA (e) descriptors using the MFCC features for the utterance $a01$ spoken by subject 03 in neutral (a) and affective (anger) (b) states.

4.4 Classification Results and Analysis

In this section, we present the classification results and analysis of our proposed emotion detection and recognition system introduced in chapter 3. We conducted various experiments to investigate the optimal number of hidden neurons in the network. We concluded that a good rule of thumb is to choose the number of hidden neurons equal to half of the number of input neurons. For each experiment, the system is tested using the 5 acoustic features described in the feature extraction section. The results of the different classifiers are combined using the two fusion techniques. The process is repeated three times for cross-validation and the results are averaged.

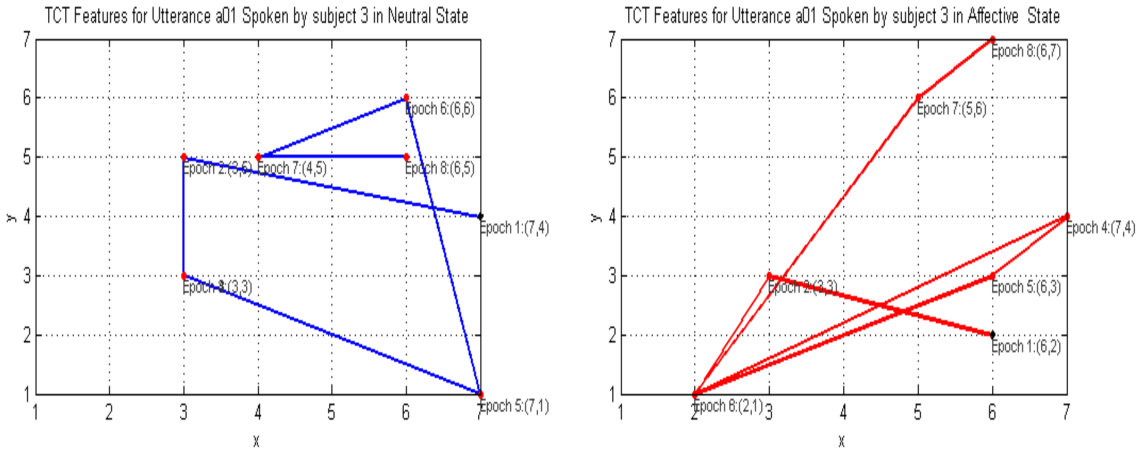


Figure 4.20: Comparison of the TCT descriptors using the MFCC features for the utterance $a01$ spoken by subject 03 in neutral (a) and anger (b) states.

Table 4.7 represent the classification results of the five feature sets for the three experiments. As it can be seen, in the first scenario, our proposed feature mapping framework outperformed consistently the SBMA descriptors. For instance, using the MFCC, the TRAA and TCT descriptors outperformed the SBMA descriptors.

For the individual features, the best accuracy rate (95.14%) is achieved by the Temporal Response Averaging Algorithm using the MFCC features. In fact, the MFCC features are efficient for detecting affective states independently of the feature mapping algorithm used.

Figure 4.21 represent the ROC curve of the TAA, TRAA, TCT, and SBMA descriptors using MFCC features and the first scenario. As it can be seen in the ROC curves, the TRAA and TCT descriptors outperform the two other algorithms namely TAA and SBMA. In fact, the Response Averaging Algorithm and Temporal Contextual Trajectory Algorithm map the input temporal sequence into a global representation while considering the discrimination between classes. On the other hand, the Temporal Averaging Algorithm and the Statistics Based Mapping Algorithm map each sequence of features (utterance) into a global representation independently of other utterances. As it can be seen, the TRAA features slightly outperform the TCT de-

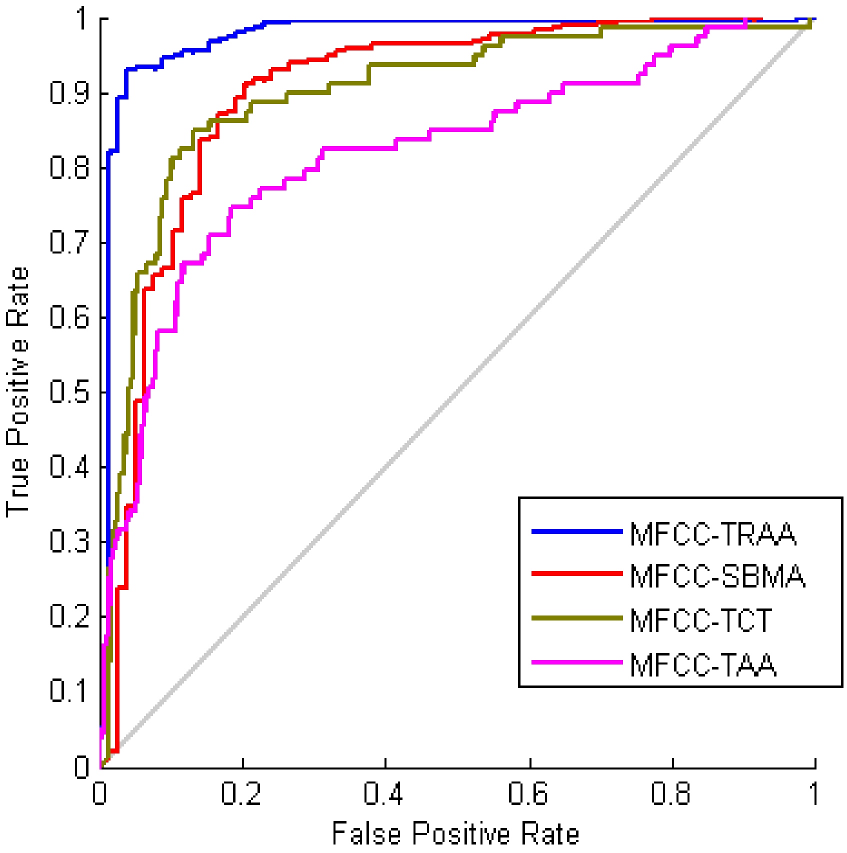


Figure 4.21: Comparison Between ROC curves of the TAA, TRAA, TCT, and SBMA descriptors using the MFCC features for the affective class.

scriptors. The same conclusions can be found using other acoustic features such as low level and PLP-RASTA.

As discussed in section 3.3.5, we use two score level fusion techniques. The first method is based on the simple averaging of the different scores. The second technique is a product based rule fusion. As it can be seen in table 4.7, the fusion of the different acoustic features improved the accuracy of the detection of the affective states independently of the method. The product rule based fusion is a severe rule. Thus, its performance is strongly correlated with the performance of the individual classifiers. As it can be seen, the performances of the product rule fusion are strongly correlated with the performances of the individual classifiers. For instance, the product fusion of the SBMA descriptors is correlated with Pitch-SBMA. On the other hand, the sum rule based fusion is more stable and less sensitive to noise and the performance of the individual classifiers. As it can be seen, the sum rule fusion improved the performances of all mapping algorithms.

In the second scenario, the proposed framework outperformed consistently the Statistics Based Mapping Algorithm. For instance, using the MFCC, the TRAA descriptors outperformed the SBMA descriptors. The same conclusions can be made using other acoustic features such as low level and PLP-RASTA. The best classification accuracy (83.72%) is achieved using the TRAA with the MFCC features. Similarly to the first scenario, the MFCC are the most efficient features for the detection of positive and negative states independently of the mapping algorithm used. The fusion of the different acoustic features improved the detection of the negative and positive states by more than than 2%. However, the sum rule based fusion provided better performances than the product rule fusion for the four different mapping algorithms.

Figure 4.22 represents the comparison between the ROC curves of the two fusion methods using the MFCC features and the Temporal Contextual Trajectory Algorithm in the detection of negative and neutral states respectively.

As it can be seen, the Sum rule score fusion outperforms the product rule score fusion in the detection of negative and neutral states. In fact, as the number of features increases, the sensitivity of the decision to the different features increases. As we increase the number of features used in the experiment, the sum rule score fusion becomes more robust to noise than the product rule and consequently more efficient.

In the third scenario, the TRAA descriptors outperformed the three other mapping algorithms. The best classification accuracy was achieved by the TRAA descriptors using the MFCC features. We conclude that the MFCC features are efficient for the detection of affective, negative, and positive emotional states. In fact the MFCC

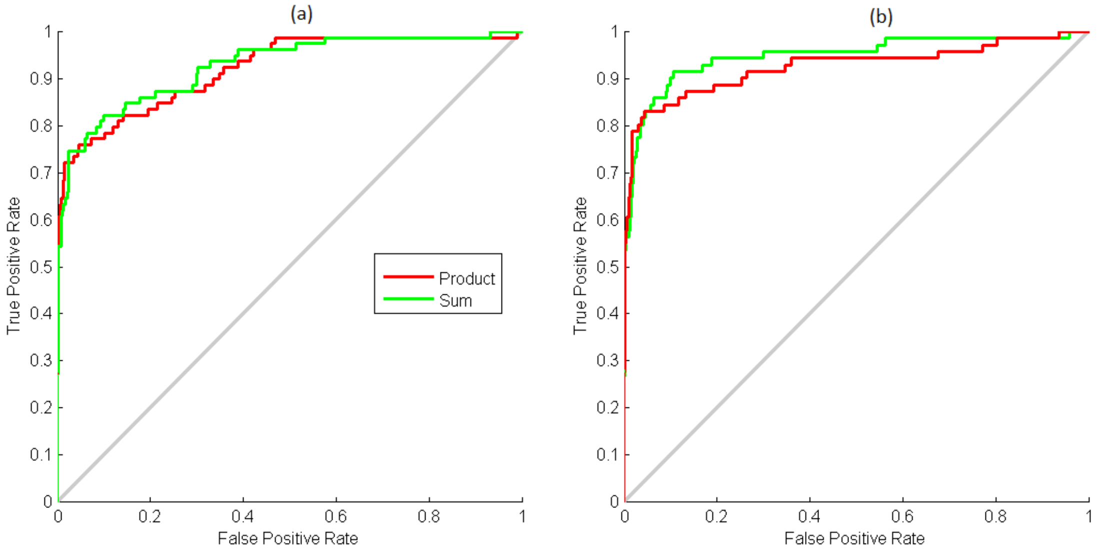


Figure 4.22: Comparison Between ROC curves of the two fusion methods using the MFCC features and the Temporal Contextual Trajectory Algorithm in the detection of negative (a) and neutral (b) states

features proved to be effective in almost all speech analysis tasks such as speech recognition, speaker segmentation and clustering, and voice activity detection. The fusion of the different acoustic features further improved the performances of our system in the recognition of human emotions such as anger and happiness. Figure 4.23 represents the ROC curve of the TRAA descriptors using the MFCC, PLP, Low Level and their Product Rule Based Fusion for the anger class (scenario 3). As it can be seen, the fusion of the different acoustic features increased the performances of our system. The curves of the MFCC and PLP are highly correlated.

From table 4.7, we can also conclude that as we increase the number of classes and the complexity of the experiments, the performances of the TRAA and TCT descriptors becomes more similar. In the first scenario, the TRAA descriptors outperformed the TCT descriptors significantly. However, in the second and the third scenario, the performances of the two descriptors became more similar.

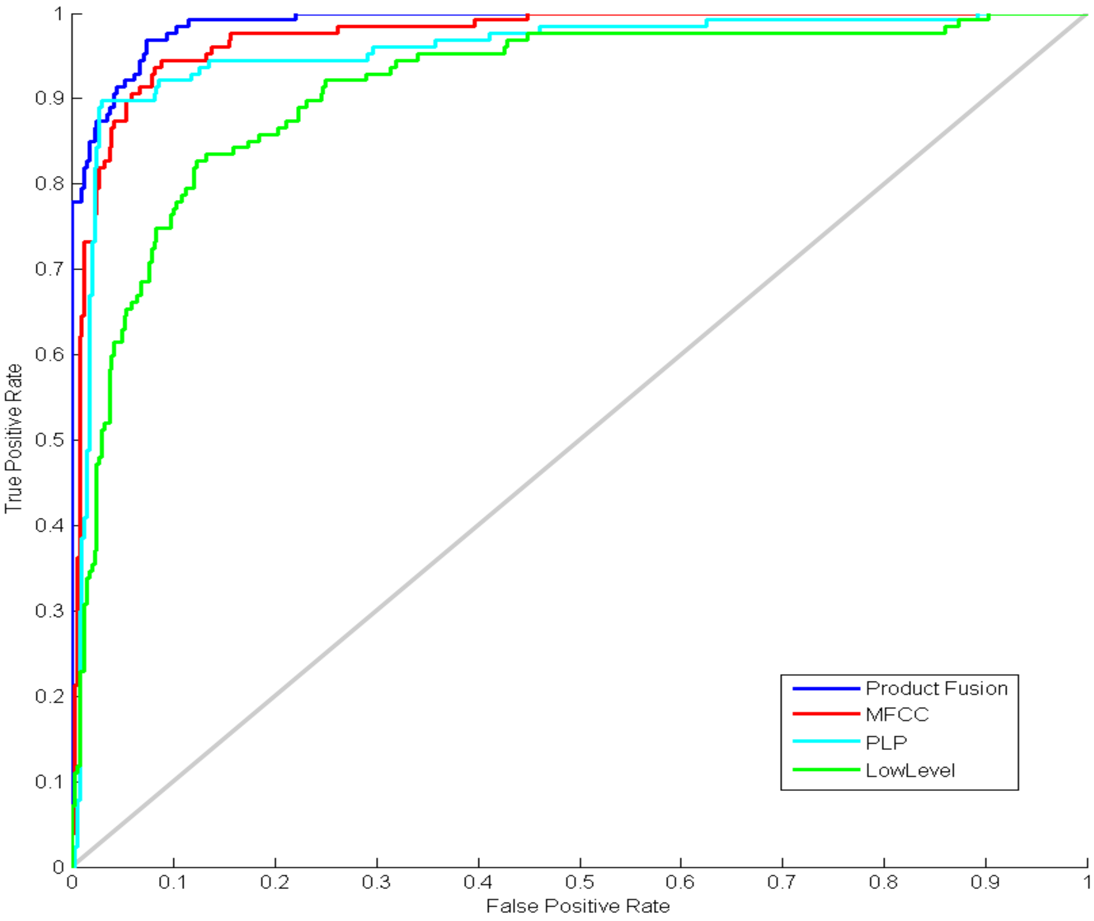


Figure 4.23: Comparison Between ROC curves of the TRAA descriptors using the MFCC, PLP, Low Level, and their Product Rule Based Fusion in the recognition of the anger emotional state

TABLE 4.7

Classification Results (%).

	Scenario 1	Scenario 2	Scenario 3
MFCC-TAA	82.24	72.34	62.80
MFCC-TRAA	**95.14**	**83.72**	**75.31**
MFCC-TCT	85.84	82.06	70.33
MFCC-SBMA	81.04	76.17	71.14
PLP-TAA	86.15	72.52	65.05
PLP-TRAA	**90.09**	**81.50**	**71.93**
PLP-TCT	86.92	77.38	64.74
PLP-SBMA	82.05	69.96	65.42
Low Level-TAA	86.17	71.59	51.40
Low Level-TRAA	**94.58**	78.13	**68.04**
Low Level-TCT	86.54	**81.12**	54.99
Low Level-SBMA	82.80	72.15	42.92
Pitch-TAA	81.23	71.96	47.85
Pitch-TRAA	**88.22**	65.05	**59.63**
Pitch-TCT	85.98	**76.64**	55.70
Pitch-SBMA	83.74	71.4	53.46
δMFCC-TAA	81.47	74.21	60.07
δMFCC-TRAA	**88.22**	**80.56**	**70.94**
δMFCC-TCT	86.31	79.54	68.41
δMFCC-SBMA	81.23	79.16	66.26
Fusion Product-TAA	84.61	73.08	64.21
Fusion Product-TRAA	**95.70**	**85.35**	**78.51**
Fusion Product-TCT	87.85	83.93	77.45
Fusion Product-SBMA	83.74	72.15	67.23
Fusion Sum-TAA	85.23	81.96	74.13
Fusion Sum-TRAA	**96.11**	**85.91**	**79.50**
Fusion Sum-TCT	88.04	84.67	78.99
Fusion Sum-SBMA	85.08	82.15	75.23

CHAPTER 5

EXPERIMENTAL RESULTS: EMOTION ANALYSIS FROM DEBATES

In the previous chapter, we used the proposed framework to detect and recognize emotions from speech using an acted dataset namely, the Berlin Emotional Dataset (BED). We showed that the proposed mapping algorithms outperform the widely used Statistics Based Mapping Algorithm while providing a better and more intuitive representation that can be used for the analysis, classification, and visualization of emotions.

In the effort to extend our work, in this chapter, we apply, report, and analyze the results of the proposed system using an authentic dataset. In this scenario, the emotional speech is naturally recorded from spontaneous people in real life situations. Such situations include customer service calls, debates, audio from video recordings in public places, and 911 calls. In this chapter, we use the proposed system to analyze and categorize human emotions from debates.

First, we outline the motivations behind the use of debates. Second, we introduce the data collection and the parameters settings used for the experiments. Third, we provide and analyze the performances of the proposed system.

5.1 Motivations

Our choice of using debates is due to various reasons.

First, debates contain typically various and complex emotions. The detection and recognition of such emotions will enable us to automatically index debates to detect emotions and confidence levels. Such scenario is useful for news channels.

Second, debates are easy to obtain through web sites such as YouTube and DailyMotion. Thus, web crawlers can be set up to download several debates that can be used in the analysis.

Third, debates are authentic. That will enable us to test our system on a real world dataset. Due to the nature of debating, the underlying emotions are usually split into affective and neutral which simplifies the learning problem for our system. Affective states in debates are usually a mixture of emotional arousals characterized by a raised tone and pitch, change in voice and speech rate, and increase in energy. We seek to detect and recognize these emotional arousals in debates.

To our knowledge, debates did not get much attention in emotion analysis from speech. In fact, there are no easily available annotated debates in the literature of emotion analysis from speech. Hence, the motivation and need to create our own corpus for the experiments.

5.2 Corpus and Parameters Settings

5.2.1 Corpus

In this chapter, we use a collection of debates to evaluate the performances of our system on a real world dataset. For that purpose, we download a collection of debates from the video platform YouTube. To ensure that the chosen debates are effective for the experiments, we search for debates using the keywords ”heated debates”.

The collection consists of 7 political debates downloaded from the video platform YouTube using the software youtube-dl [80].

The audio information used in our experiments was extracted form the video using

Audio Online Converter [81]. The political debates have variable length, emotions, number and gender of speakers in order to maximize the different scenarios in the experiments. There are 3 nationalities namely British, American, and Indian and 3 different accents in the used debates.

For each debate, we manually divided the debate into segments to include only one speaker and one emotion per segment. For a more automatic process, speaker segmentation and clustering can be used to divide the audio recordings prior emotion analysis. In our experiments, we used the software Audacity [82] to divide the debates into segments. All the mentioned tools are freely available on the internet. The segments have variable length ranging from 2 seconds to 6 seconds. As discussed in the third chapter, our proposed framework maps any temporal sequence into the same representation which alleviates the issue with dealing with variable length time series data. As mentioned previously, emotions in debates are rather neutral or affective due to the nature of debating. Hence, we adopt the first scenario of the previous chapter in which the goal is to detect affective emotions from neutral.

After dividing the debates, we manually labelled all the segments with the underlying emotion of the speaker. The details of the collection is displayed in table 5.8

TABLE 5.8

Details of The Collection of Political Debates.

Number	Name	YouTube ID	Number of Speakers	Number of Segments
1	Gun Laws Debate	RC4JJWUtzkc	2	19
2	Michael Moore on CNN	2JMCryfTtTI	3	14
3	Republicans 2012 Debate	c37VcgHUFVk	3	35
4	Debate on ESPN	j6x-O3kb1sI	3	18
5	Obama and Romney 2012 Debate	NXkLYIZabWE	3	18
6	British Immigration Debate	6ZdQ0kA3ksg	3	11
7	Indian News Debate(English)	bAQq2mENhR4	2	37
	Total		19	152

5.2.2 Parameters Settings

5.2.2.1 Frame and Hop Size

Similarly to the previous experiments in chapter 4, we use the hope and frame size. That is, a window size of 40 ms with 50% overlap (that is 20 ms as the hop size). The same window and hop size are used throughout this chapter. Our goal is to generalize the same parameters settings for different datasets and applications.

5.2.2.2 Training and Learning

The collections of segments from the different debates are all combined then divided into training, testing, and validation sets. The training set is used to estimate the parameters of the mapping and the learning model. The testing set is used to test the system. The validation set is used by the neural network to validate the learning model of the classifier. The different segments from all debates are combined into one collection and divided into 70%, 15%, and 15% for training, testing, and validation respectively. The process is repeated three times for cross-validation.

5.3 Feature Extraction

In our experiments, we use the same features from the previous chapter. These features are MFCC, ΔMFCC, Pitch (F0), Perceptual linear predictive (PLP) Rasta , and Low Level Spectral and Temporal Features. For a detailed description of the acoustic features used in this experiment, please refer to section 4.2 and Appendix A.

5.4 Feature Mapping

In this section, we analyze the performances of the proposed framework using the collection of political debates. Recalling the same methodology from the previous chapter, each audio segment is mapped into descriptors using the proposed feature mapping framework. We refer to the global descriptor computed by the Temporal Averaging Algorithm by TAA Descriptors. We refer to the global descriptor computed by the Temporal Response Averaging Algorithm by TRAA descriptors. We refer to the global descriptor computed by the Temporal Contextual Trajectory Algorithm by TCT descriptors.

Figure 5.24 and 5.25 represent respectively the comparison of the TAA and TRAA descriptors of two segments expressing affective and neutral states using the MFCC features. The segments are spoken by the same speaker and taken from debate 1. The memory parameter w is set to 0.05 for the TAA and TRAA descriptors. The SOMs map size is set to 7×7 for the TRAA descriptors. The response for the TRAA descriptors is computed in a crisp fashion. For each segment, we compute the TAA (1×13), TRAA (1×49) descriptors as described in the third chapter.

As it can be seen, both the TAA and TRAA descriptors were able to capture the difference in emotions between the two segments. Using the TAA descriptors, we can notice the difference in coefficients 5 to 8. In fact, the MFCC coefficients represent the power of short term spectrum computed on the mel-scale frequency. In our case, we computed 13 MFCC coefficients. In the case of affective states, we expect those value to vary in specific frequency ranges due to activity such as increase in the tone or yelling which are captured in specific frequencies. In the case of neutral states, we expect the values to vary in other frequency ranges.

In the TRAA descriptors, the two segments have different distributions on the codebook of the SOMs. In fact, the map units of the SOMs are clusters that are labeled and represent specific emotions. High responses to map units that are labelled as affective suggest emotional activity. Similarly, high responses to map units that are labelled as neutral suggest no emotional activity. Highly mixed clusters typically rep-

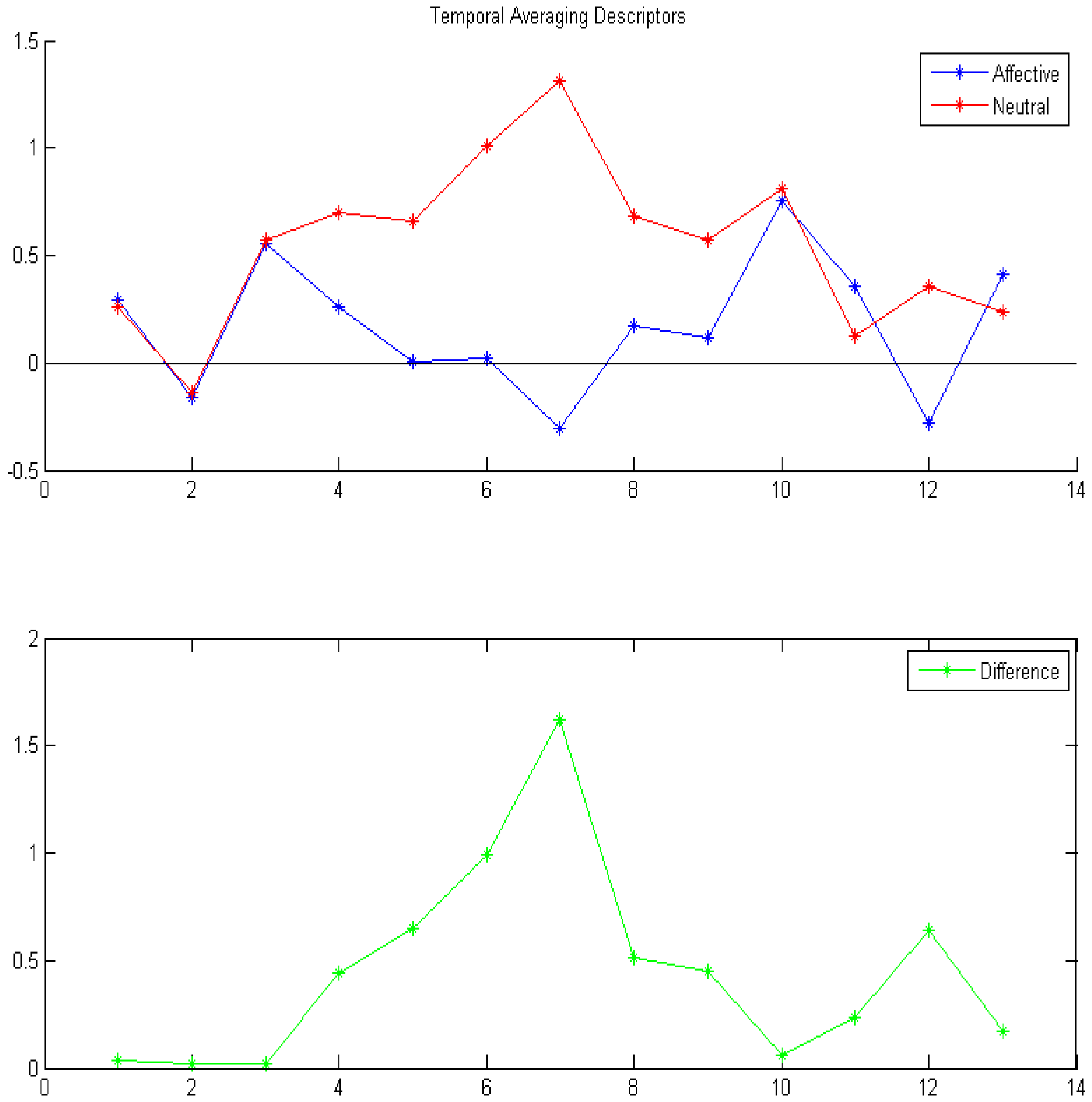


Figure 5.24: Comparison of the TAA descriptors using the MFCC features for two segments expressing affective and neutral states spoken by the same speaker. (Debate 1)

resent the overlap between the two classes. Such correlations are learned by Neural Network during training and evaluation.

Figure 5.26 displays the comparison of the TCT descriptors of the two segments from the previous example. As it can be seen, the two trajectory occupy different regions of the map. In the case of the neutral segment, the trajectory is mainly located in the upper area of the map. On the other hand, the affective trajectory occupy the

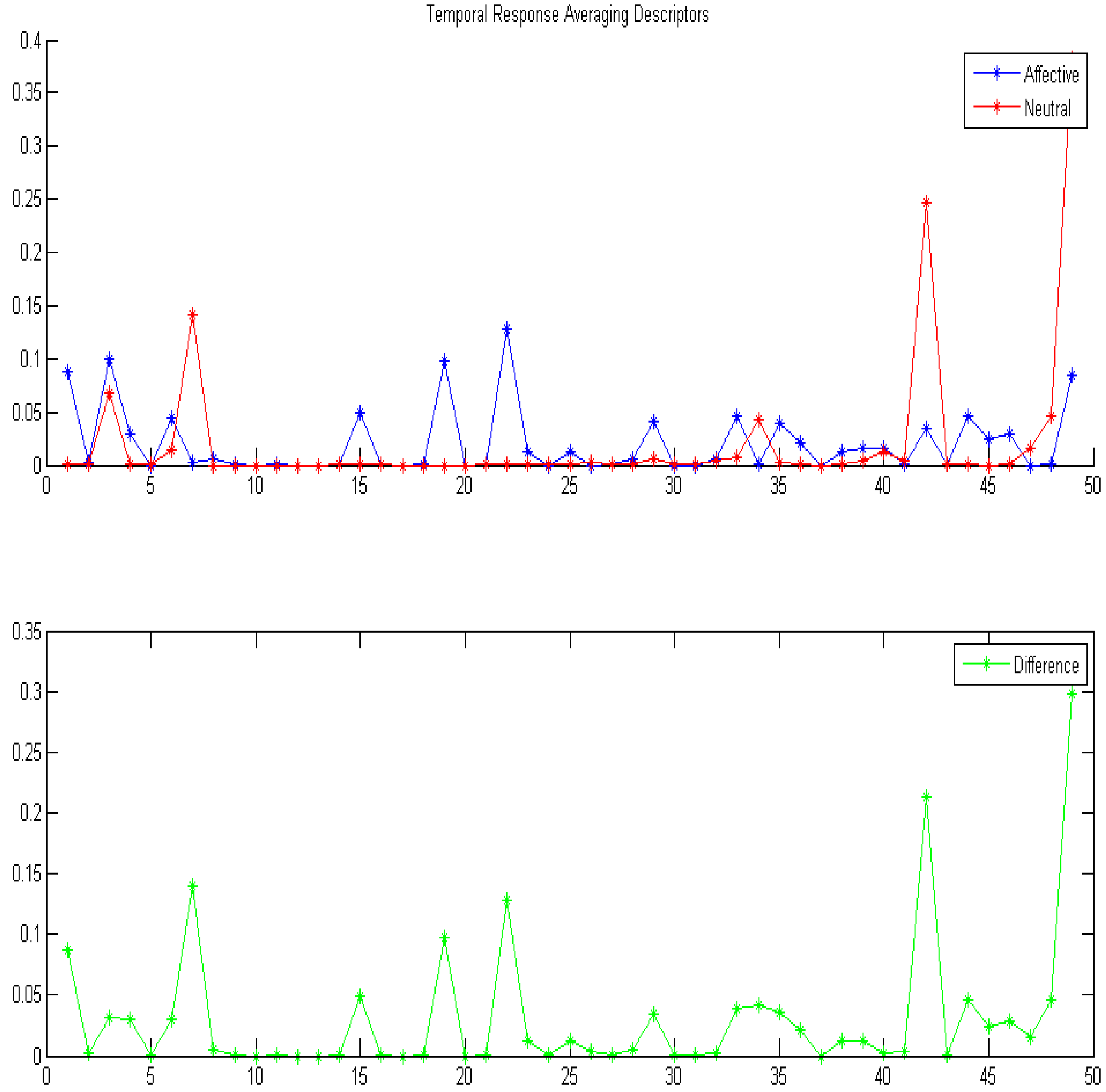


Figure 5.25: Comparison of the TRAA descriptors using the MFCC features for two segments expressing affective and neutral states. (Debate 1)

lower part of the map. Due to topology preserving of the SOMs, nearby map units are similar. Thus, map units labelled as affective are grouped together occupying specific regions of the map. Similarly, the neutral map units are located on different sides of the map. Typically, the class borders are located in the middle of the map. Thus, map units in the middle of the map are typically highly mixed.

The two segments in the previous example are spoken by the same speaker. Figure

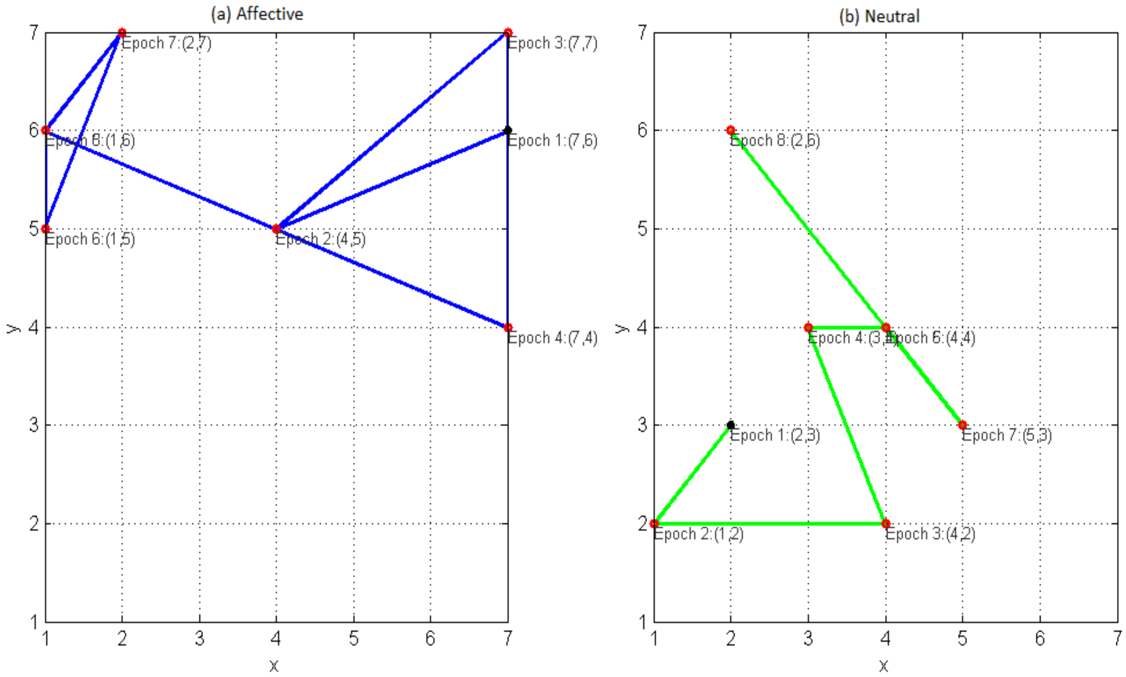


Figure 5.26: Comparison of the TCT descriptors using the MFCC features for two segments expressing affective and neutral states spoken by the same speaker. (Debate 1)

5.27 and 5.28 represent respectively the comparison of the TAA and TRAA descriptors of two segments expressing affective and neutral states respectively using the MFCC features spoken by two different speakers. The segments are taken from Debate 1 and 2 respectively. The same parameters from the previous example are used to compute the TAA and TRAA descriptors.

In figure 5.27, the TAA descriptors of the two segments are correlated for some coefficients such as 5,9, and 12 and dissimilar in other coefficients such as 10 and 11. Similarly, in figure 5.28, the TRAA descriptors of the two segments have different distributions on the codebook.

We compute the pairwise distance of the TCT descriptors of all segments in the collection using a simple Euclidean distance. Then, we identify the closest neighbors of each segment in the collection. Figure 5.29 represents the TCT descriptors of an affective segment and its closest neighbor using the Euclidean distance. The two seg-

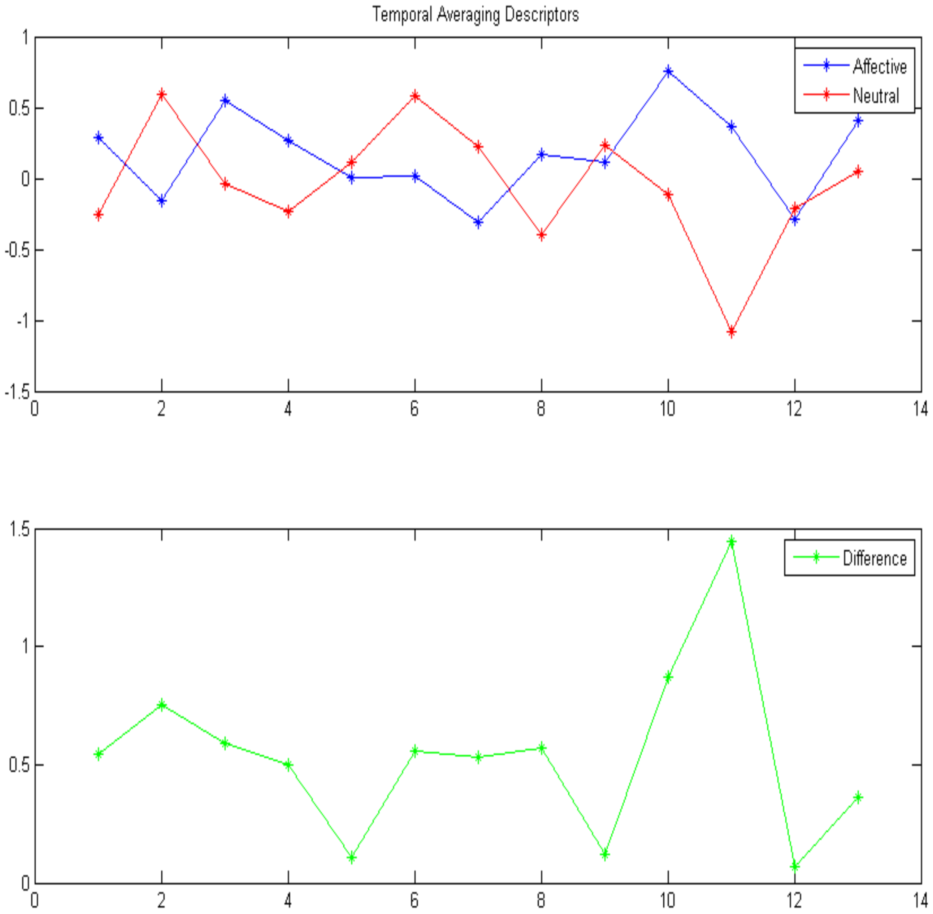


Figure 5.27: Comparison of the TAA descriptors using the MFCC features for two segments expressing affective and neutral states spoken by two different speakers. (Debate 1 and 2)

ments are spoken by two different speakers from two different debates (Debate 1 and 7 respectively). The true label of the closest segment is also affective. In fact, the first segment is spoken by a male and its closest is spoken by a female. Moreover, the two segments are spoken in Indian English and British English respectively. As it can be seen, the two trajectories exert the same behavior on the map. Despite being from different debates, the TCT descriptors are able to capture the similarity in emotional activity between the two audio segments. Since we are using the MFCC, the units represent energy clusters. The TCT descriptors tracks the fluctuations of the energy

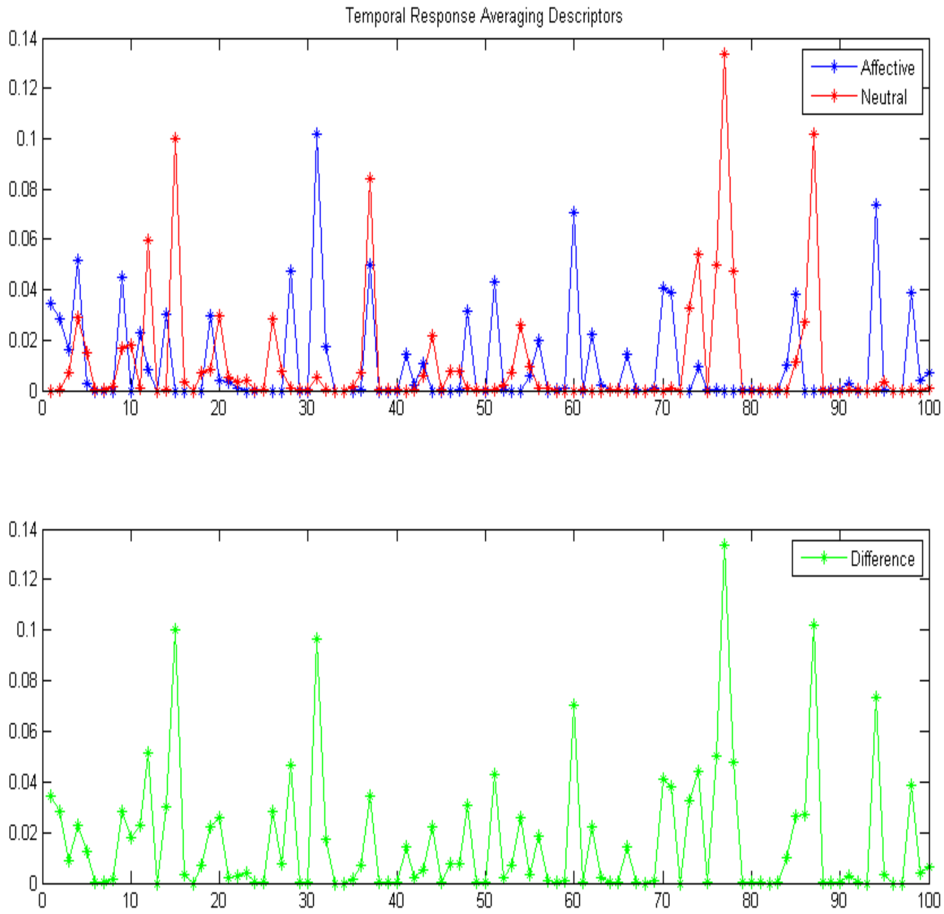


Figure 5.28: Comparison of the TRAA descriptors using the MFCC features for two segments expressing affective and neutral states spoken by two different speakers. (Debate 1 and 2)

level over time.

5.5 Classification Results and Analysis

In this section, we present the classification results and analysis of our proposed emotion detection and recognition system applied on the collection of debates introduced in section 5.2.1. After mapping, the descriptors are divided into training, testing, and validation. For each experiment, the system is tested using the 5 acoustic

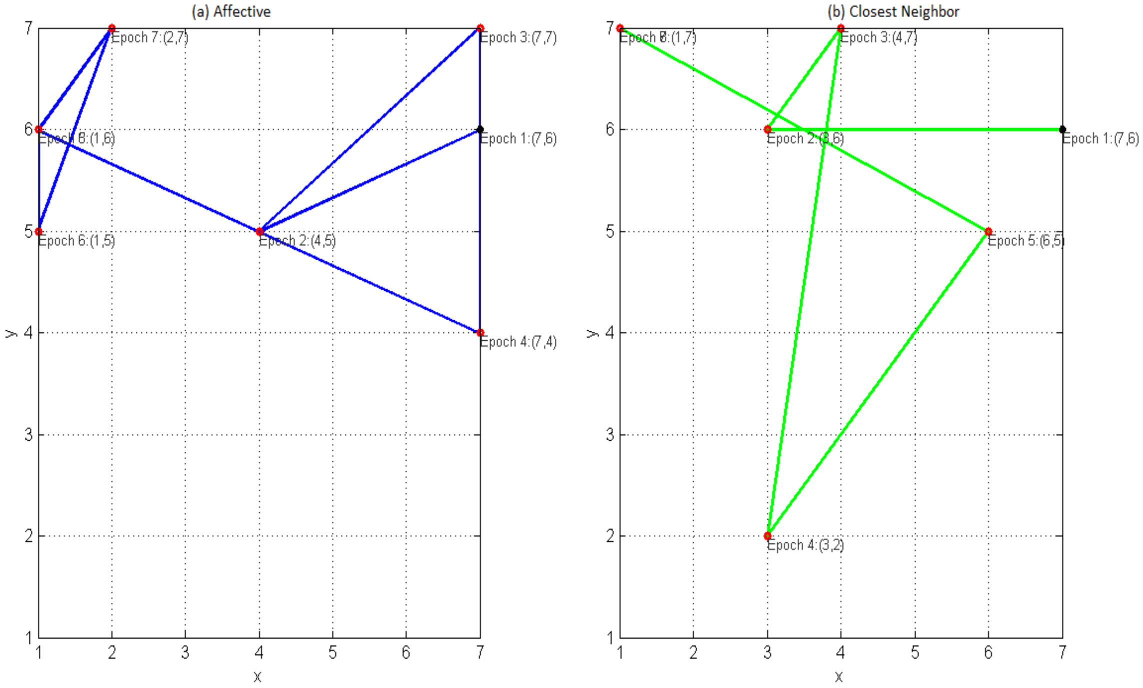


Figure 5.29: Comparison of the TCT descriptors using the MFCC features for two segments expressing affective (a) and its closest neighbor(b). (Debate 1 and 7)

features described in the feature extraction section. We use a different classifier for each acoustic feature, resulting into 5 neutral networks. We use half of the input size as the number of hidden neurons. The scores of the different classifiers are combined using the two proposed fusion techniques. The first one is a product rule fusion. The second approach is based on summation. This process is repeated three times and the classification results are averaged. Table 5.9 represent the classification results of the five feature sets for the three experiments.

As it can be seen, the discrimination power of the proposed system decreased significantly compared to the performances on the acted corpus in the previous chapter. This is due to many reasons. First, the debate collection contains different audio qualities. On the other hand, the Berlin Emotional Dataset has the same quality for all utterances. Second, the number of speakers in the debate collection is higher than the Berlin Emotional Dataset. In fact, there are 19 speakers in the debate collections

TABLE 5.9

Classification Results (%).

	Debates	BED (scenario 1)
MFCC-TAA	50.78	82.24
MFCC-TRAA	69.57	95.14
MFCC-TCT	**73.91**	85.84
PLP-TAA	56.52	86.15
PLP-TRAA	60.87	90.09
PLP-TCT	**69.57**	86.92
Low Level-TAA	52.17	86.17
Low Level-TRAA	56.52	94.58
Low Level-TCT	**62.17**	86.54
Pitch-TAA	60.87	81.23
Pitch-TRAA	55.22	88.22
Pitch-TCT	**56.52**	85.98
δMFCC-TAA	56.14	81.47
δMFCC-TRAA	60.87	88.22
δMFCC-TCT	**69.57**	86.31
Fusion Product-TAA	65.22	84.61
Fusion Product-TRAA	**78.26**	95.70
Fusion Product-TCT	75.22	87.85
Fusion Sum-TAA	73.91	85.23
Fusion Sum-TRAA	78.26	96.11
Fusion Sum-TCT	**78.94**	88.04

compared to 10 subjects in the BED dataset. The Berlin Emotional Dataset is also repetitive since the same sentences are used throughout the corpus where the debates are very versatile in vocabulary. Third, there are 3 nationalities and 3 accents in the debate collection compared to one language and one accent in the Berlin Emotional Dataset. Moreover, the emotions in debates are typically mixed and more complex compared to acted emotions. Thus, the training data must be big enough to capture these variations. Larger training can be used in these experiments, however, it is time and effort consuming to manually label a large collection of debates. All these limitations and challenges contribute in the deterioration of the system's performances. Despite these limitations, the proposed system is still able to provide competitive results. For instance, using the TCT descriptors and the MFCC features, the system was able to average 73.91% accuracy rate. Using the other acoustic features, the TCT descriptors are able to average around 70% accuracy rate. In these experiments, we used only 7 debates. In order to improve the accuracy rate, the size of the data and the number of debates must increase.

Figure 5.30 represents the ROC curve the TAA, TRAA, and TCT descriptors using the MFCC features. As it can be seen, the TCT and TRAA both outperformed the

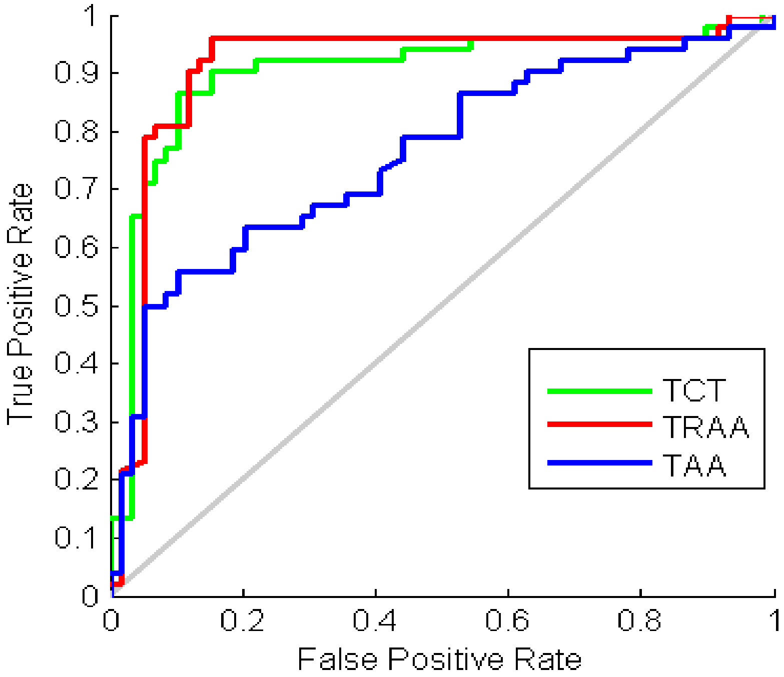


Figure 5.30: Comparison of the ROC curves of the TAA, TRAA, and the TCT descriptors using the MFCC features for the affective class.

TAA descriptors due to their data driven nature. Both the TCT and TRAA descriptors provide similar and competitive results using the MFCC features. However, the

TCT slightly outperformed the TRAA descriptors. This suggests that the TCT descriptors are more robust to noise and the configuration of the corpus. In the previous chapter, the TRAA descriptors outperformed the TCT descriptors in the first scenario but in the second and third scenario the performances of the two descriptors became similar. In fact, the summarization method proposed for the Temporal Contextual Trajectory is more robust to noise than the Temporal Response Averaging Algorithm. As pointed out in the third chapter, the center of mass is stable and robust to noise. In the Temporal Response Averaging Algorithm, each map unit is mapped to its best match unit with a certain response. In the case of real world datasets, noise is very common and equally contributes in the computation of the response on the map and deteriorates its performances. In order to improve the performances of the TRAA descriptors, we must use a response computation that is robust to noise.

The summation rule fusion provided better results than the product rule fusion. In fact, the sum based rule improved the classification accuracy of the system independently of the mapping algorithm used. For instance, using the TCT features, the sum rule fusion improved the performances of the system by 5%. On the other hand, the product based rule increased the performances of the system in the case of the TCT descriptors by only less than 2%. As pointed out in the previous chapter, product fusion is a severe rule fusion and as the number of features and the complexity of the emotions increase, the fusion score becomes correlated with the score of the individual classifiers. Similarly to the previous chapter, the pitch features perform the worst out of the 5 acoustic features. Figure 5.31 displays the comparison between the ROC curves of the two fusion techniques using the 5 different acoustic features and the TCT descriptors. As it can be seen, the sum fusion outperforms the product fusion in the detection of neutral states. For the affective class, the results of the two fusion techniques are highly correlated.

In order to test further the generalization power of our system in cross-corpus emotion detection and recognition, we conduct the following experiment. First, we train the proposed system using the collection of debates. Then, we download a new debate and use it to test our system. Thus, the training data do not contain any occurrence of the test debate. Similarly, to the collections of debates, the debate is preprocessed and segmented into speaker and emotion homogenous segments. Next, 5 acoustic features are extracted and their representation is mapped using the mapping parameters from the training data. Finally, the resulting descriptors are fed to the Neural Network. The classification step is repeated three times and the classification results are averaged. Table 5.10 represents the details of the test debate. We divide the debate into 69 segments of three speakers. The accents of the three speakers is quiet different also from the ones in training data.

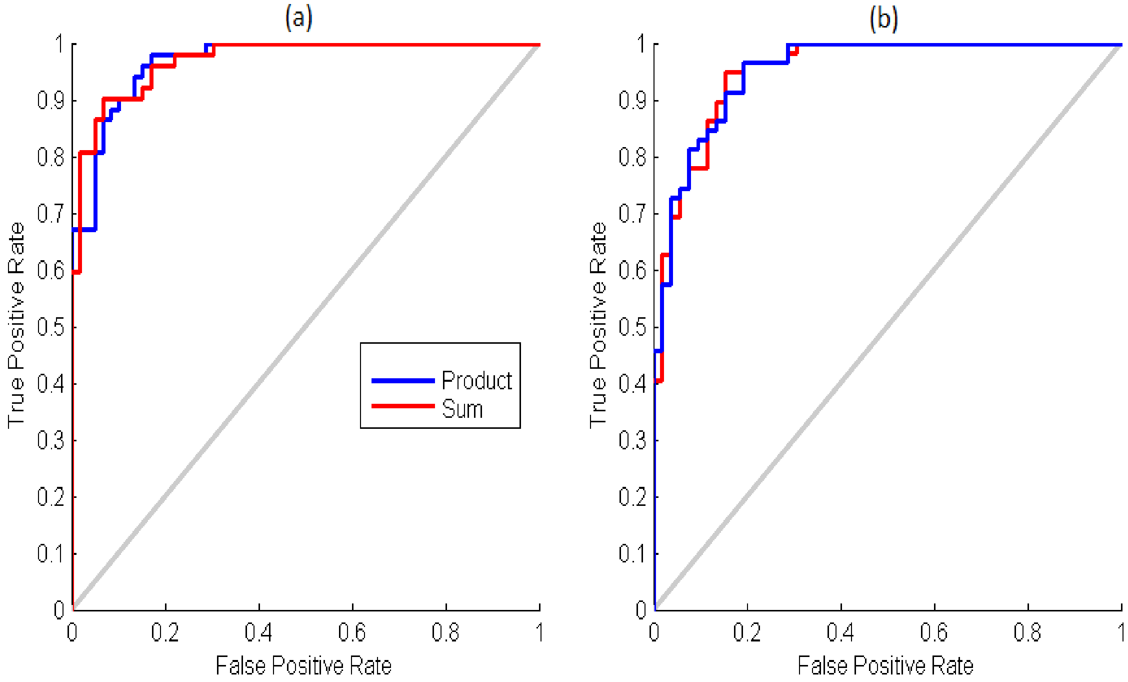


Figure 5.31: Comparison of the ROC curves of the two fusion techniques using the 5 different acoustic features and the TCT descriptors for the neutral (a) and affective (b) class.

Table 5.11 displays comparison of the performances of the proposed system on the

TABLE 5.10

Details of The Test Debate.

Number	Name	YouTube ID	Number of Speakers	Number of Segments
1	NFL Debate	RC4JJWUtzkc	3	69

Berlin Emotional Dataset, The debate collection, and the test debate. For simplicity, we display only the final fusion results. As it can be seen, despite the deterioration in the performances, our system is still able to average 70% accuracy on an unforseen debate. Similarly, the Temporal Contextual Trajectory Algorithm outperformed the TAA and TRAA. This suggests, that the TCT descriptors integrates the discrimination between emotions more efficiently and have better generalization power. The TCT descriptors are also more robust to noise than the TRAA and TAA descriptors. The performances of the TRAA and TAA also suggest that the temporal averaging step is not robust to noise.

TABLE 5.11

Classification Results (%).

	Test Debate	Debates	BED
Fusion Product-TAA	53.48	65.22	84.61
Fusion Product-TRAA	62.32	**78.26**	95.70
Fusion Product-TCT	**66.67**	75.22	87.85
Fusion Sum-TAA	54.93	73.91	85.23
Fusion Sum-TRAA	60.87	78.26	96.11
Fusion Sum-TCT	**71.01**	**78.91**	88.04

Figure 5.32 represents the comparison of the ROC curves of the TCT algorithms using the Berlin Emotional Dataset, the Debate Collection, and the test debate. As it can be seen, the TCT descriptors are still able to generalize to unforseen data with competitive performances.

Emotion analysis from debates can be further improved in many ways. First, the training debate data can be increased to include more situations, emotions, and configurations. As the size of the training data increases, the mapping increases its ability to discriminate between emotions in mapping and classification.

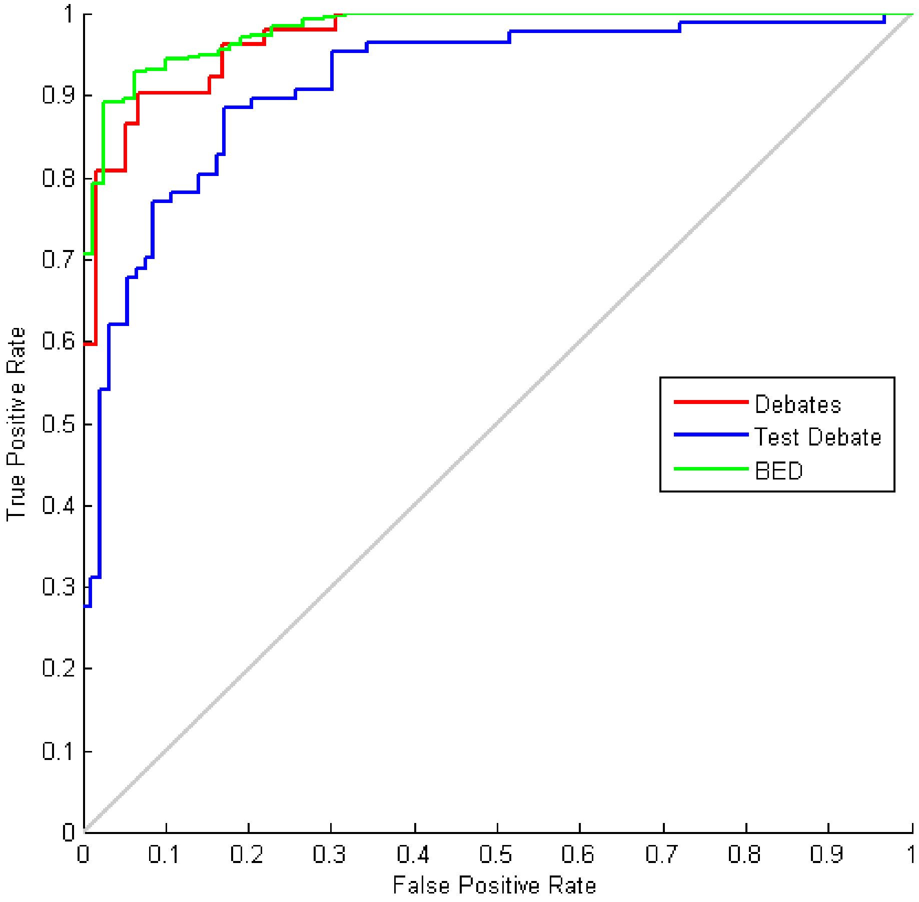


Figure 5.32: Comparison of the ROC curves of the proposed system TCT descriptors for the the detection of neutral states using sum fusion and the three experiments.

Second, in the case of emotion detection and recognition from debates, speaker segmentation and clustering can be used prior to emotion analysis to detect and recognize speakers in the analysis. Combining emotion analysis and speaker segmentation and clustering will enable us to fully index political debates to detect and recognize automatically the number of speakers involved in the conversation, then, categorize their emotions in debates. Such systems can be further improved by using speech recognition to transcript the entire debate. Combining the proposed descriptors with text descriptors will further improve the overall performances of our system.

Due to the nature of debating, at the feature level, we can improve the system by creating features that are specific to measure emotions in debates. Such features can measure changes in the tone, pitch, and energy.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis, we have presented a novel framework for mapping emotional speech data into global descriptors that integrates the temporal information from the original sequence and the discrimination between emotions.

In most classic approaches, temporal speech data is mapped into a static global descriptor before analysis and classification. This often results in the loss of temporal ordering from the original sequence. Emotion is the result of a succession of acoustic events. By discarding the temporal ordering of these events in the mapping, the classic approaches cannot detect acoustic patterns that lead to a certain emotion. The proposed framework overcomes these limitations by integrating the temporal information in the mapping implicitly and explicitly. Moreover, the proposed framework is data driven since it integrates the discrimination between emotions in the mapping using unsupervised learning.

The proposed framework includes three mapping algorithms. In the first algorithm, the Temporal Averaging Algorithm, the data is averaged using leaky integrators to produce a global descriptor that preserve some of the temporal information in the original sequence. The temporal information is integrated implicitly in the descriptor. In order to integrate the discrimination between classes in the mapping, we proposed the Temporal Response Averaging Algorithm which combines the temporal averaging step of the previous algorithm and unsupervised learning to produce data driven temporal contextual descriptors.

The third algorithm, the Temporal Contextual Trajectory Algorithm, maps a temporal sequence into an ordered trajectory representing the behavior over time of the input utterance on a 2-D map of emotions. This was achieved using the topology preserving property of the Self Organizing Maps and the continuous nature of speech. The temporal information is integrated explicitly in the decriptor which makes it easier to monitor emotions in long speeches.

The proposed mapping framework map speech data of different length to the same equivalent representation which alleviates the problem of dealing with variable length temporal sequences. This is advantageous in real time setting where the size of the analysis window is variable.

In order to test the framework's performances, we proposed a novel emotion detection and recognition system that adopts our mapping framework. The goal of such system is to index emotional databases to facilitate the classification and retrieval of emotions. The proposed system was applied on two emotional datasets.

The first dataset is an acted dataset. We have shown that the proposed framework outpeforms the widely used Statistics Based Mapping Algorithm while provinidng a better and more intuitive representation that can be used for the analysis, classification, and visualization of emotions.

The second dataset is an authentic dataset. In this thesis, we used the proposed system to index debates in order to detect and monitor human emotions in long debate conversations. For that purpose, we have created and labelled a collection of debates that can be used in emotion analysis. Such initiative is one of the first in the emotion analysis from speech literature. We showed that the proposed emotion detection and recognition system provides competetive results on the debate dataset with different speakers, accents, and configurations.

The performance of the different mapping algorithms depends mainly on the acoustic features and the dataset. According to our experiments, the Temporal Response Averaging Algorithm and the Temporal Contextual Trajectory Algorithm outpeforms the Temporal Averaging Algorithm due to their data driven nature. We also concluded that the Temporal Contextual Trajectory Algorithm achieves better generalization in cross-corpus settings than the Temporal Response Averaging Algorithm. This is due to the fact that the proposed summarization method for the TCT descriptors is more robust to noice than the temporal repsonse averaging.

The performances of the different features and algorithms is further improved using socre level fusion. In this thesis, we derived, used and compared two score level fusion techniques. The first one is a product based rule. The second technique is based on summation. We showed the fusion of the individual classifiers improves the overall performances of the system on the two different datasets. We also showed that the sum based fusion is more robust in noisy enviroments than the product rule fusion.

6.2 Future Work

Although the proposed framework have shown comptetive results, there is still room for improvement.

For instance, in the first two porposed mapping algorithms (TAA and TRAA), the temporal averaging step was performed using leaky integrators. Future research may include investigating other temporal averaging techniques that more robust to noise. The discrimnation between emotions was integrated in the mapping using the Self Organizing Maps. In this thesis, we used the classic version of the SOMs. Another possible approach is to use other variants of the SOMs. One option is the Growing Self Organizing Maps [83, 84]. The GSOM aims to overcome the issue of fixing apriori the size of the map by automatically identifying from the data. This will. enable us to reduce the number of parameters governing our system.

In the Temporal Contextual Trajectory Algorithm, the sub trajectories are chosen as consecutive non overlapping windows. Future research may include using other dynamic window decomposition techniques. One possible approach is to decompose the trajectory into variable length subtrajectories where the behavior is uniform. Another approach is to use overlapping subtrajectories in order to eliminate discontinuities in the summarized trajectories.

The proposed debate corpus contains currently 8 debates. Future work may include adding new debates to increase the size of the training data.

The proposed framework maps any sequential data into temporal contextual descriptors that integrate the order of the original data.

Future work may include investigating the use of the proposed framework in other applications that use other type of sequential data such as financial data (stock market).

REFERENCES

[1] Nicu Sebe, Ira Cohen, Theo Gevers, and Thomas S. Huang, "Multimodal approaches for emotion recognition: a survey," *Proc. SPIE*, vol. 5670, pp. 56–67, 2005.

[2] Murray Grossman Nii Martey John Bell Mark Liberman, Kelly Davis, "Emotional prosody speech and transcripts," 2002.

[3] SreenivasaRao Krothapalli and ShashidharG. Koolagudi, "Speech emotion recognition: A review," in *Emotion Recognition using Speech Features*, SpringerBriefs in Electrical and Computer Engineering, pp. 15–34. Springer New York, 2013.

[4] B. Schuller, G. Rigoll, and M. Lang, "Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 2004, vol. 1, pp. I–577–80 vol.1.

[5] ShashidharG. Koolagudi and K.Sreenivasa Rao, "Emotion recognition from speech: a review," *International Journal of Speech Technology*, vol. 15, no. 2, pp. 99–117, 2012.

[6] SreenivasaRao Krothapalli and ShashidharG. Koolagudi, "Characterization and recognition of emotions from speech using excitation source information," *International Journal of Speech Technology*, vol. 16, no. 2, pp. 181–201, 2013.

[7] Ralph Adolphs, "Recognizing emotion from facial expressions: Psychological and neurological mechanisms," 2002.

[8] R.A. Patil, V. Sahula, and A. S. Mandal, "Automatic recognition of facial expressions in image sequences: A review," in *Industrial and Information Systems (ICIIS), 2010 International Conference on*, 2010, pp. 408–413.

[9] A.M. Adeshina, Siong-Hoe Lau, and Chu-Kiong Loo, "Real-time facial expression recognitions: A review," in *Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009*, 2009, pp. 375–378.

[10] Jia Rong, Gang Li, and Yi-Ping Phoebe Chen, "Acoustic feature selection for automatic emotion recognition from speech," *Inf. Process. Manage.*, vol. 45, no. 3, pp. 315–328, May 2009.

[11] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *in Proceedings of Interspeech, Lissabon*, 2005, pp. 1517–1520.

[12] Dimitrios Ververidis and Constantine Kotropoulos, "A review of emotional speech databases," .

[13] J. Kangas, "Phoneme recognition using time-dependent versions of self-organizing maps," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 101–104 vol.1, 1991.

[14] About.com, "Theories of Emotion," `http://psychology.about.com/od/psychologytopics/a/theories-of-emotion.htm`, 2014, [Online; accessed 12-December-2014].

[15] Prof. M. B. Zalte Dipti D. Joshi, "Speech emotion recognition: A review," *IOSR Journal of Electronics and Communication Engineering*, 2013.

[16] M. Schrder, "Emotional speech synthesis: A review," *Proceedings of the 7th european conference on speech communication and technology, 2nd interspeech event*, 2001.

[17] Iain R. Murray, John L. Arnott, and Elizabeth A. Rohwer, "Emotional stress in synthetic speech: progress and future directions," *Speech Communication*, vol. 20, no. 1-2, pp. 85–91, Nov. 1996.

[18] Ying Wang, Shoufu Du, and Yongzhao Zhan, "Adaptive and optimal classification of speech emotion recognition," in *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, 2008, vol. 5, pp. 407–411.

[19] Iker Luengo, Eva Navas, Inmaculada Hernez, and Jon Snchez, "Automatic emotion recognition using prosodic parameters," *in Proc. of INTERSPEECH*, pp. 493–496, 2005.

[20] S. Ramakrishnan, "Recognition of emotion from speech: A review," *Speech Enhancement, Modeling and Recognition Algorithms and Applications*, 2001.

[21] M. Wollmer, F. Eyben, B. Schuller, Ellen Douglas-Cowie, and Roddy Cowie, "Data-driven clustering in emotional space for affect recognition using discriminatively trained lstm networks," *10th INTERSPEECH Conference*, pp. 1555–1558, 2009.

[22] V. Sethu, E. Ambikairajah, and J. Epps, "Speaker normalisation for speech-based emotion detection," in *Digital Signal Processing, 2007 15th International Conference on*, 2007, pp. 611–614.

[23] F.N. Julia and K.M. Iftekharuddin, "Detection of emotional expressions in speech," in *Proceedings of the IEEE SoutheastCon, 2006.*, 2006, pp. 307–312.

[24] Richard M. Stern and Alejandro Acero, "Acoustical pre-processing for robust speech recognition," in *Proceedings of the workshop on Speech and Natural Language*, Stroudsburg, PA, USA, 1989, HLT '89, pp. 311–318, Association for Computational Linguistics.

[25] L. Rabiner, M. Cheng, A.E. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.

[26] Onur Babacan, Thomas Drugman, Nicolas d'Alessandro, Nathalie Henrich, and Thierry Dutoit, "A comparative study of pitch extraction algorithms on a large variety of singing sounds," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7815–7819.

[27] Patricio De La Cuadra and Aaron Master, "Efficient pitch detection techniques for interactive music," in *Proceedings of the 2001 International Computer Music Conference, La Habana*, 2001.

[28] Oxford Dictionnaries, "Definition of Pitch at Oxford Dictionnaries," `http://www.oxforddictionaries.com/us/definition/american_english/pitch`, 2014, [Online; accessed 12-December-2014].

[29] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 5, pp. 293 – 302, jul 2002.

[30] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[31] David H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[32] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.

[33] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and Bernhard Scholkopf, "An introduction to kernel-based learning algorithms," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 181–201, 2001.

[34] ShashidharG. Koolagudi, Sudhamay Maity, VuppalaAnil Kumar, Saswat Chakrabarti, and K.Sreenivasa Rao, "Iitkgp-sesc: Speech database for emotion analysis," in *Contemporary Computing*, Sanjay Ranka, Srinivas Aluru, Rajkumar Buyya, Yeh-Ching Chung, Sumeet Dua, Ananth Grama, SandeepK.S. Gupta, Rajeev Kumar, and VirV. Phoha, Eds., vol. 40 of *Communications in Computer and Information Science*, pp. 485–492. Springer Berlin Heidelberg, 2009.

[35] Dimitrios Ververidis and Constantine Kotropoulos, "A state of the art review on emotional speech databases," .

[36] B. S. Atal, "Automatic speaker recognition based on pitch contours," *The Journal of the Acoustical Society of America*, vol. 52, no. 6B, pp. 1687–1697, 1972.

[37] H. Wakita, "Residual energy of linear prediction applied to vowel and speaker recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 3, pp. 270–271, 1976.

[38] Philippe Thevenaz and Heinz Hugli, "Usefulness of the lpc-residue in text-independent speaker verification," *Speech Communaction*, vol. 17, no. 1-2, pp. 145–157, Aug. 1995.

[39] J.W. Cooley and J.W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comp.*, pp. 297–301, 1965.

[40] Ali N. Akansu and Richard A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, Academic Press, Inc., Orlando, FL, USA, 1992.

[41] Tsang-Long Pao, Yu-Te Chen, Jun-Heng Yeh, and Wen-Yuan Liao, "Combining acoustic features for improved emotion recognition in mandarin speech," in *Affective Computing and Intelligent Interaction*, vol. 3784 of *Lecture Notes in Computer Science*, pp. 279–285. 2005.

[42] C. E. Williams and K. N. Stevens, "Vocal correlates of emotional states," in *Speech Evaluation in Psychiatry*, 1981, pp. 219–224.

[43] Kornel Laskowski Daniel Neiberg, Kjell Elenius, "Emotion recognition in spontaneous speech using gmms," *In proceedings of the 9th ISCA International Conference on Spoken Language Processing*, 2006.

[44] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, 2007, pp. 3–24, IOS Press.

[45] Zeng Fan-Zi and Qiu Zheng-Ding, "A survey of classification learning algorithm," in *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, 2004, vol. 2, pp. 1500–1504 vol.2.

[46] David D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," *Proceedings of the 10th European Conference on Machine Learning*, pp. 4–15, 1998.

[47] Laveen N. Kanal, "Perceptron," in *Encyclopedia of Computer Science*, pp. 1383–1385. John Wiley and Sons Ltd., Chichester, UK.

[48] Sebastian Mika, Gunnar Rtsch, Jason Weston, Bernhard Schlkopf, and Klaus-Robert Mller, "Fisher discriminant analysis with kernels," *Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pp. 41–48, 1999.

[49] L. Satish and B. I. Gururaj, "Use of hidden markov models for partial discharge pattern classification," *Electrical Insulation, IEEE Transactions on*, vol. 28, no. 2, pp. 172–182, 1993.

[50] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[51] Ivo D. Dinov, "Expectation maximization and mixture modeling tutorial," *UCLA: Statistics Online Computational*, 2008.

[52] Jeff Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Tech. Rep., International Computer Science Institute, 1998.

[53] R. ; Bangham J.A. Gibson, S. Harvey, "Multi-dimensional histogram comparison via scale trees," *Image Processing Proceedings*, pp. 709–712 vol.2, 2001.

[54] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 3, pp. 418–435, May 1992.

[55] Haythem Balti and Hichem Frigui, "Feature mapping and fusion for music genre classification.," in *ICMLA*. 2012, pp. 306–310, IEEE.

[56] Adel S. Elmaghraby Haythem Balti, "Speech emotion detection using time dependent self organizing maps," in *IEEE International Symposium on Signal Processing and Information Technology*, 2013.

[57] S. Planet and I. Iriondo, "Comparison between decision-level and feature-level fusion of acoustic and linguistic features for spontaneous emotion recognition," in *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, June 2012, pp. 1–6.

[58] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern Analysis and Applications*, vol. 1, no. 1, pp. 18–27, 1998.

[59] J. Kittler, A. Hojjatoleslami, and T. Windeatt, "Strategies for combining classifiers employing shared and distinct pattern representations," *Pattern Recogn. Lett.*, vol. 18, no. 11-13, pp. 1373–1377, Nov. 1997.

[60] A. Sayedelahl, R. Araujo, and M.S. Kamel, "Audio-visual feature-decision level fusion for spontaneous emotion estimation in speech conversations," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, July 2013, pp. 1–6.

[61] Merijn van Erp and Lambert Schomaker, "Variants of the borda count method for combining ranked classifier hypotheses," in *IN THE SEVENTH INTERNATIONAL WORKSHOP ON FRONTIERS IN HANDWRITING RECOGNITION. 2000. AMSTERDAM LEARNING METHODOLOGY INSPIRED BY HUMANS INTELLIGENCE BO ZHANG, DAYONG DING, AND LING ZHANG*, 2000, pp. 443–452.

[62] Pinaki. Chowdhury, Sukhendu. Das, Suranjana. Samanta, and Utthara. Mangai, "A Survey of Decision Fusion and Feature Fusion Strategies for Pattern Classification," 2010, vol. 27, pp. 293–307.

[63] Niklas Lavesson, Niklas Lavesson, Niklas Lavesson, and Niklas Lavesson, "Evaluation and analysis of supervised learning algorithms and classifiers," .

[64] J. Kangas, "Time-delayed self-organizing maps," *International Joint Conference on Neural Networks*, pp. 331–336 vol.2, 1990.

[65] K. Torkkola and M. Kokkonen, "Using the topology-preserving properties of sofms in speech recognition," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 261–264 vol. 1, 1991.

[66] Adel S. Elmaghraby Haythem Balti, “Emotion analysis from speech using temporal contextual trajectories,” in *IEEE International Symposium on Computers and Communications*, 2014.

[67] D.R.W. Barr, P. Dudek, J.M. Chambers, and K. Gurney, “Implementation of multi-layer leaky integrator networks on a cellular processor array,” in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, Aug 2007, pp. 1560–1565.

[68] J.G. Taylor, “Temporal patterns and leaky integrator neurons,” in *International Neural Network Conference*, pp. 952–955. Springer Netherlands, 1990.

[69] T. Kohonen, *Self-organization and associative memory: 3rd edition*, Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[70] Michael Biehl, Anarta Ghosh, Barbara Hammer, and Yoshua Bengio, “Dynamics and generalization ability of lvq algorithms,” in *Journal of Machine Learning Research*, 2006.

[71] R.P. Lippmann, “An introduction to computing with neural nets,” *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, 1987.

[72] Jing Li, Ji-hang Cheng, Jing-yuan Shi, and Fei Huang, “Brief introduction of back propagation (bp) neural network algorithm and its improvement,” in *Advances in Computer Science and Information Engineering*, vol. 169 of *Advances in Intelligent and Soft Computing*, pp. 553–558. Springer Berlin Heidelberg, 2012.

[73] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis, “Comparative evaluation of various mfcc implementations on the speaker verification task,” in *Proc. of the SPECOM-2005*, 2005, pp. 191–194.

[74] Slaney M., *Auditory Toolbox. Version 2*, Interval Research Corporation, 1998.

[75] S. Kim, P.G. Georgiou, Sungbok Lee, and S. Narayanan, “Real-time emotion detection system using speech: Multi-modal fusion of different timescale features,” in *IEEE 9th Workshop on Multimedia Signal Processing*, 2007, pp. 48–51.

[76] Wei Chu and Abeer Alwan, “Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend,” in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Washington, DC, USA, 2009, ICASSP ’09, pp. 3969–3972, IEEE Computer Society.

[77] Hynek Hermansky, “Perceptual linear predective (plp) analysis of speech,” *Acoustic Society*.

[78] H. Hermansky and N. Morgan, “Rasta processing of speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[79] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, “Rasta-plp speech analysis technique,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing.*, 1992, vol. 1, pp. 121–124 vol.1.

[80] “Youtube-dl: http://rg3.github.io/youtube-dl/,” .

[81] “Audio online converter: http://audio.online-convert.com/convert-to-mp3,” .

[82] “Audacity: http://audacity.sourceforge.net/,” .

[83] A. Rauber, D. Merkl, and M. Dittenbach, “The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 6, pp. 1331–1341, Nov 2002.

[84] Junlin Zhou and Yan Fu, “Clustering high-dimensional data using growing som,” in *Advances in Neural Networks ISNN 2005*, Jun Wang, Xiao-Feng Liao, and Zhang Yi, Eds., vol. 3497 of *Lecture Notes in Computer Science*, pp. 63–68. Springer Berlin Heidelberg, 2005.

[85] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[86] H.E. Rickard, G.D. Tourassi, and A.S. Elmaghraby, “Self-organizing maps for masking mammography images,” *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, pp. 302–305, 2003.

[87] H.E. Rickard, G.D. Tourassi, N. Eltonsy, and A.S. Elmaghraby, “Breast segmentation in screening mammograms using multiscale analysis and self-organizing maps,” in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, 2004, vol. 1, pp. 1786–1789.

APPENDIX **A**

In this appendix, we provide mathematical equations and details of implementation of various acoustic feature extraction, signal processing algorithms, and the Self Organizing maps.

A.1 LP Residual Estimation Algorithm

All the computations performed in this section are performed at the frame level. Our goal is to compute the LP residual e of an audio frame $\mathbf{x}$ of length N.
The redundancy in the speech signal is exploited in the LP analysis. That is, the current sample is a linear combination of past p samples where p denotes the order of prediction.
Thus, the predicted sample $\hat{\mathbf{x}}$ is defined as follow

$$\hat{\mathbf{x}(n)} = -\sum_{k=1}^{p} a_k.\mathbf{x}(n-k) \tag{36}$$

In (36), $\{a_k, 1 \leq k \leq p\}$ are the linear prediction coefficients.
The prediction error (LP residual) e is defined as the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$, that is

$$e(n) = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x}(n) + \sum_{k=1}^{p} a_k.\mathbf{x}(n-k) \tag{37}$$

The primary goal of LP analysis is to compute the LP coefficients $\{a_k, 1 \leq k \leq p\}$ which minimized the prediction error e. Various methods have been proposed in the literature to compute the LP coefficients such as the covariance method and the autocorrelation method[38, 6]. In this proposal, we use the least squares auto correlation method which seeks to minimize the total predication error defined as

$$E = \sum_{n=-\infty}^{+\infty} e^2(n) \tag{38}$$

We replace e by its expression in equation (37),

$$E = \sum_{n=-\infty}^{+\infty} \{\mathbf{x}(n) + \sum_{k=1}^{p} a_k.\mathbf{x}(n-k)\}^2 \tag{39}$$

The LP coefficients $\{a_k, 1 \leq k \leq p\}$ which minimize the total prediction error E are the solution of the following equation

$$\frac{dE}{da_k} = 0, 1 \leq k \leq p \tag{40}$$

$$\frac{dE}{da_k} = \frac{d}{da_k}\{\sum_{n=-\infty}^{+\infty} \{\mathbf{x}(n) + \sum_{k=1}^{p} a_k.\mathbf{x}(n-k)\}^2\} = 0, 1 \leq k \leq p \tag{41}$$

Since

$$\sum_{n=-\infty}^{+\infty} \{\mathbf{x}(n-j).\mathbf{x}(n)\} = \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{+\infty} \{\mathbf{x}(n-j).\mathbf{x}(n-k)\} \tag{42}$$

Equation (42) can be rewritten in terms of an autocorrelation sequence R as follow

$$\sum_{k=1}^{p} a_k.R(j-k) = R(j) \tag{43}$$

Where the autocorrelation function R of length N is defined as follow

$$R(j) = \sum_{n=j}^{N-1} \mathbf{x}(n).\mathbf{x}(n-j) \tag{44}$$

Equation (44) can be written in the matrix form as follow

$$R.A = -r \tag{45}$$

In (45), R is a pxp symmetric matrix of elements $R(i,k) = R(|i-k|), 1 \leq i,k \leq p$, $r = \{R(1), R(2), ..., R(p)\}$ is a column vector. $A = \{a_1, a_2, .., a_p\}$ is the column vector of LP coefficients. R is toeplitz matrix and thus invertible. Thus, A is defined as follow

$$A = -R.r \tag{46}$$

A.2 Feature Extraction Algorithm of Low Level Features

All the computations in this section are performed at the frame level. Let x denotes an input audio frame of size N.

A.2.1 Spectral Centroid

The spectral centroid is defined as the center of gravity of the magnitude spectrum of the short term fourier transform (STFT). The Spectral Centroid of frame $\mathbf{x}$ is computed as follow:

$$C = \frac{\sum_{j=1}^{J} P(j) \times j}{\sum_{j=1}^{J} P(j)} \tag{47}$$

In (47), $P(j)$ denotes the power spectrum of $\mathbf{x}$ (equation **??**)at frequency bin j. The centroid is a measure of spectral shape and higher centroid values correspond to brighter textures with more high frequencies.

A.2.2 Spectral Roll-Off

The spectral roll-off is defined as the frequency R below which 85% of the magnitude distribution is concentrated. The Spectral roll-off of frame $\mathbf{x}$ is computed as follow:

$$\sum_{j=1}^{R} P(j) = 0.85 \times \sum_{j=1}^{J} P(j) \tag{48}$$

In (48), $P(j)$ denotes the power spectrum of $\mathbf{x}$ (equation **??**)at frequency bin j. The spectral roll-off is another measure of spectral shape.

A.2.3 Spectral Flux

The spectral flux is defined as the squared difference between the normalized magnitudes of successive spectral distributions. The Spectral flux of frame $\mathbf{x}$ is computed as follow:

$$F = \sum_{j=1}^{J} (N_i[j] - N_{i-1}[j])^2 \tag{49}$$

In (49), N_i and N_{i-1} denotes the normalized power spectrum at the current frame i, and the previous frame $i-1$, respectively.

The spectral flux is a measure of the amount of local spectral change.

A.2.4 Zero Cross Rate

The zero cross rate is defined as the rate of sign-changes along a signal. The zero cross rate of frame $\mathbf{x}$ is computed as follow:

$$Z = \frac{1}{2} \sum_{n=1}^{N} |sign(\mathbf{x}[n]) - sign(\mathbf{x}[n-1])| \tag{50}$$

Time domain zero crossings provide a measure of the noisiness of the signal.

A.2.5 Short Time Energy

The Short time energy of frame $\mathbf{x}$ is computed as follow:

$$E = \frac{1}{N} \sum_{j=1}^{N} |\mathbf{x}(j)|^2 \tag{51}$$

A.3 Self Organizing Feature Maps

Self-organizing maps (SOMs)[85, 86, 87, 69] is a class of artificial neural network based on competitive learning. It is widely used for data clustering and visualization. The SOMs maps in the input data samples on a map of usually 1 or 2 dimensions which plot the similarities of the data by grouping similar data items together. Thus, SOMs accomplish two things, it reduces the dimension of the input data and displays similarities between data on a 2-D map (clustering). During mapping, the topology of the input space is preserved on the map by using a topological neighborhood function. The 2-D map is defined by its map units. Each map unit is defined by its location on the map and a prototype vector.

The SOMs can be trained in batch or sequential mode. In this section, we outline the sequential version of the SOMs.

Let $X = \{X_i, 1 \leq i \leq N\}$ denotes the input data to the SOMs. Let u_i denotes the i^{th} map unit of the SOMs of size $M \times M$ and let $\mathbf{C}_i$ denotes its corresponding prototype vector of dimension d.

The sequential SOMs have five steps.

1. Initialization: Initialize the prototype vectors C. Various initialization methods have been proposed in the literature such as random, and linear. In the random initialization, a set of M^2 data samples are chosen randomly from the input data.

2. Sampling: A data sample $\mathbf{X}_j$ is chosen randomly from X.

3. Matching: The distance between $\mathbf{X}_j$ and all map units is computed. The best matching unit of $\mathbf{X}_j$ is defined as

$$||\mathbf{X}_j - C_{bmu}|| = \min_i\{||\mathbf{X}_j - C_i||\} \tag{52}$$

4. Updating: The prototype of the best matching unit $\mathbf{C}_{bmu}$ and its topological neighbors are moved closer to the input vector in the input space. The update rule for the prototype vector of unit i is defined as

$$\mathbf{C}_i^{(t+1)} = \mathbf{C}_i^{(t)} + \alpha(t)h_{bi}(t)[\mathbf{X}_j - \mathbf{C}_i(t)] \tag{53}$$

In (53), t denotes the t^{th} iteration, $\alpha(t)$ is an adaptation coefficient, and $h_{bi}(t)$ is a neighborhood kernel centered on the winner unit typically defined as follow

$$h_{bi}^{(t)} = \exp(-\frac{||r_b - r_i||}{||2(\sigma^(t))^2||}) \tag{54}$$

In (54), r_b and r_i are positions of neurons b and i on the SOM grid. Both $\alpha^{(t)}$ and $\sigma^{(t)}$ decrease monotonically with time.

5. Convergence: Repeat steps 2-5 until α reaches 0.

CURRICULUM VITAE

Haythem Balti, Doctoral Candidate

- Address: 1511 S. 3rd St. Apt. 5 Louisville, KY 40208
- Phone: (502) 235-2920
- Email: haythembalti@gmail.com

EDUCATION

- 2009-2014 : Ph.D. University of Louisville
  Computer Science and Engineering
- 2006-2009 : B.Eng. Higher School of Telecommunications.
  Network and Telecommunications Engineering
- 2004-2006 : Associate Degree. Preparatory School of Bizerte.
  Maths and Physics

TEACHING AND RESEARCH EXPERIENCE

- August 2013-August 2014 : University of Louisville Graduate Teaching Assistant, Louisville, KY.
- August 2009-December 2014 : University of Louisville Graduate Research Assistant, Louisville, KY.

PROFESSIONAL EXPERIENCE

- July 2014- Present Software Developer, White Clay, Louisville, KY

- August 2012-August 2013 Software Developer Intern, GMeals LCC, Louisville, KY

- August 2011-August 2012 Web Developer Intern, Pixowl LLC, Louisville, KY

RESEARCH PAPERS AND PRESENTATIONS

- Balti, H.; Elmaghraby, A.S., "Speech emotion detection using time dependent self organizing maps," Signal Processing and Information Technology(ISSPIT), 2013 IEEE International Symposium, pp.000470,000478, 12-15 Dec. 2013

- Balti, H.; Elmaghraby, A.S., "Emotion analysis from speech using temporal contextual trajectories," Computers and Communication (ISCC), 2014 IEEE Symposium, pp.1,7, 23-26 June 2014