UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

On Automatic Emotion Classification Using Acoustic Features

by

Ali Hassan

Thesis for the degree of Doctor of Philosophy

June 22, 2012

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

Doctor of Philosophy

ON AUTOMATIC EMOTION CLASSIFICATION USING ACOUSTIC FEATURES

by Ali Hassan

In this thesis, we describe extensive experiments on the classification of emotions from speech using acoustic features. This area of research has important applications in human computer interaction. We have thoroughly reviewed the current literature and present our results on some of the contemporary emotional speech databases.

The principal focus is on creating a large set of acoustic features, descriptive of different emotional states and finding methods for selecting a subset of best performing features by using feature selection methods. In this thesis we have looked at several traditional feature selection methods and propose a novel scheme which employs a preferential Borda voting strategy for ranking features. The comparative results show that our proposed scheme can strike a balance between accurate but computationally intensive wrapper methods and less accurate but computationally less intensive filter methods for feature selection.

By using the selected features, several schemes for extending the binary classifiers to multiclass classification are tested. Some of these classifiers form serial combinations of binary classifiers while others use a hierarchical structure to perform this task. We describe a new hierarchical classification scheme, which we call Data-Driven Dimensional Emotion Classification (3DEC), whose decision hierarchy is based on non-metric multi-dimensional scaling (NMDS) of the data. This method of creating a hierarchical structure for the classification of emotion classes gives significant improvements over other methods tested.

The NMDS representation of emotional speech data can be interpreted in terms of the well-known valence-arousal model of emotion. We find that this model does not give a particularly good fit to the data: although the *arousal* dimension can be identified easily, *valence* is not well represented in the transformed data. From the recognition results on these two dimensions, we conclude that *valence* and *arousal* dimensions are not orthogonal to each other.

In the last part of this thesis, we deal with the very difficult but important topic of improving the generalisation capabilities of speech emotion recognition (SER) systems over different speakers and recording environments. This topic has been generally overlooked in the current research in this area. First we try the traditional methods used in automatic speech recognition (ASR) systems for improving the generalisation of SER in intra– and inter–database emotion classification. These traditional methods do improve the average accuracy of the emotion classifier.

In this thesis, we identify these differences in the training and test data, due to speakers and acoustic environments, as a covariate shift. This shift is minimised by using importance weighting algorithms from the emerging field of *transfer learning* to guide the learning algorithm towards that training data which gives better representation of testing data. Our results show that importance weighting algorithms can be used to minimise the differences between the training and testing data. We also test the effectiveness of importance weighting algorithms on inter–database and cross-lingual emotion recognition. From these results, we draw conclusions about the universal nature of emotions across different languages.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Data

$x(n)$	Continuous input speech sample
$x[n]$	Discretised $x(n)$
x_i	Input patterns $x_i \in \mathcal{X}$
$x_i(j)$	j^{th} component of input pattern x_i
y_i	Classes labels of input x_i
$\mathcal{X}$	Input domain
n	Total number of data examples
D	Total number of features, dimension of $\mathcal{X}$
d	Number of features in a subset where $d \leq D$
M	Total number of classes
s	Total number of speakers in a database
f_0	Pitch
f_s	Sampling frequency
T_0	Time period
F_i	Centre frequency of i^{th} formant
BW_i	Bandwidth of i^{th} formant frequency
Rc	Ranking criterion
L	Total number of rankers
τ_{max}	Maximum lag

Vectors, Matrices and Norms

$\mathbf{1}$	Vector with all entries equal to one
$\mathbf{0}$	Vector with all entries equal to zero
$\boldsymbol{I}$	Identity matrix
A^{-1}	Inverse matrix
$\langle x, x' \rangle$	Dot product between x and x'
$\|\|.\|\|$	2-norm, $\|\|x\|\| := \sqrt{\langle x, x' \rangle}$

SVM-related

w	Normal vector
C	SVM misclassification tolerance parameter
b	Constant offset
α_i	Lagrange multipliers
ξ_i	Slack variables

IW-related

x_i^{tr}	Input training patterns $x_i^{tr} \in \mathcal{X}$
x_i^{te}	Input testing patterns $x_i^{te} \in \mathcal{X}$
y_i^{tr}	Training data classes $y_i^{tr} \in [M]$
y_i^{te}	Testing data classes $y_i^{te} \in [M]$
n_{tr}	Number of training examples
n_{te}	Number of testing examples
β	Importance weights
λ	Regularisation parameter

Abbreviations and Acronyms

3DEC	Data-Driven Dimensional Emotion Classification
k-NN	k Nearest Neightbour
AA	Average Accuracy
ACF	Autocorrelation Function
AMDF	Average magnitude difference function
ASC	Autism Spectrum Disorders
ASR	Automatic Speech Recognition
CMN	Cepstral Mean Normalisation
DET	Detection Error Trade-off
EER	Equal Error Rate
EER	Equal Error Rate
GMM	Gaussian Mixture Models
HCI	Human Computer Interaction
HMM	Hidden Markov Models
IW	Importance Weighting
KDE	Kernel Density Estimation
KLIEP	Kullback–Leibler importance estimation
KMM	Kernel Mean Matching
LCV	Likelihood Cross Validation
LLD	Low Level Descriptor
LPA	Linear Predictive Analysis
LPC	Linear Predictive Coding
MAP	Maximum-a-posteriori
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Logistic Regression
MLP	Multi-Layer Perceptron
MMD	Maximum mean Discrepancy
NMDS	Non-metric Multi-Dimensional Scaling
PCA	Principle Component Analysis
QP	Quadratic Programming
SD–CV	Speaker Dependent Cross Validation
SI–CV	Speaker Independent Cross Validation

SBE	Sequential Backward Elimination
SER	Speech Emotion Recognition
SFS	Sequential Forwarding Selection
SOA	State of the Art
SVM	Support Vector Machines
UA	Unweighted AA
uLSIF	Un-constrained Least Square Importance Function
VTLN	Vocal Tract Length Normalisation
WA	Weighted AA

Declaration of Authorship

I, Ali Hassan , declare that the thesis entitled *On Automatic Emotion Classification Using Acoustic Features* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: (Hassan and Damper, 2009), (Hassan and Damper, 2010), (Hassan and Damper, 2012), (Hassan *et al.*, 2012a) and (Hassan *et al.*, 2012b).

Signed:.....

Date:.....

Acknowledgements

First and foremost, my heart felt gratitude and appreciation is addressed to my supervisor Professor Bob Damper for taking me as a research student under his valuable supervision. His advice, discussions and guidance were the real encouragement to complete this work. I admire his simplicity, generosity and work ethics. It has been an honour to work with him. I will always be thankful to him for the valuable times that he spent in supervising my progress, and the research opportunities he gave me during this time. I would also like to thank Professor Mahesan Niranjan who has spent time with me discussing various aspects of this thesis. I am really grateful to him for providing me great insight into some of the difficulties that I had during my PhD. I would also like to thank Dr. Sasan Mahmoodi who monitored my progress as a member of my MPhil/PhD committee at the University of Southampton. I am thankful to National University of Scieces and Technology, Pakistan for funding first two years PhD research.

Being at the University of Southampton has been a great fun. I have been lucky to have such great friends who have given me so much love and support. I am really grateful to Ali Gondal, Basam, Banafshe, Fasih, Ke, Mohammad Tayarani, Mohsin, Musi, Ramanan, Umar Khan and Wei who have made this experience such a great pleasure and fun.

Nothing would have been possible without the support of my wife who took care of our children during my studies. I really appreciate her support for standing with me during the tough times that we have faced during this period. I have to thank my parents for their blessings and love without which none of this would have been possible.

Thank you all.

To my parents, wife and lovely daughters...

Chapter 1

Introduction

Imagine you are driving on a highway, your in-car satellite navigation system (satnav) tells you to '*take the exit at next junction*' but you know that this is the wrong exit. This will not take you through the shortest route to your destination. The exit that you really want is the one after, but your satnav keeps on reminding you *'take the next exit' ... '200 metres to the exit' ... '100 metres to the exit' ... 'turn left and take the exit' ..... 're-calculating the shortest route to your destination'*. How many times have you been in such situations and lost your temper by shouting at your satnav, although you know that it can not understand what you are saying. What if it could understand that you are loosing your patience, and adapt its further instruction accordingly or even try to sooth you by saying something nice? Similarly, what if your car could understand when you are feeling tired or frustrated with traffic, and caution you before you cause any road rage?

Such automated machines which can understand the current emotional state of their users and respond to them have been part of fiction films for some time. As an example, consider the film 'I, Robot' (released 2004), which is set in the year 2035 in which there is one robot for every five people in the world. The idea is to create robotic assistants who can understand the physical and emotional requirements of human beings and respond accordingly. In this film, Dr. Alfred Lanning, a scientist at U.S. Robotics, creates a robot named 'Sonny' which other than its usual tasks, can also understand and feel emotions. He even gets sentimental about dying.

There is another film titled 'Stealth' (released 2005), in which an unmanned combat aerial vehicle is introduced in the U.S. Navy flown by 'EDI', a computer with an artificial intelligence, alongside other fighting planes. At one stage in the film, the officials believe that EDI has become a liability and decide to destroy it by ordering it to fly to a remote Alaskan base. EDI recognises the stress patterns in the commander's voice and decides to take evasive action.

Over the last decade, this research into developing such systems which can recognise the current emotional state of the speaker have shifted from fiction to a major research topic in human computer interaction (HCI) and speech processing. It came into the focus of pattern recognition and machine learning communities with the publication of Picard's influential book *Affective Computing* (Picard, 1997). The aim of this research is to develop speech emotion recognition (SER) systems that can not only understand the verbal contents, but also emotions in the speech to which any human listener would understand and react. This research has a lot of applications in spoken dialogue systems such as computer enhanced learning systems, automated response systems, etc. However, there are still many issues that need to be addressed before this technology becomes part of our daily life.

1.1 Current Issues in SER Systems

Darwin (1859) had argued that emotions are survival-related responses that have evolved to solve certain problems that all species have faced over their evolutionary lifecycle. Consequently, emotions have to be, in general, the same in all human beings. In addition, because of the common evolutionary past of humans and other animals, similarities in the emotions should be observable for closely-related species. Any social organism has means of communicating their emotions to their species. Each uses different means to communicate their current state ranging from secreting chemicals to using body gestures or language. If we take an example of ants, they secrete chemicals to communicate their messages to other ants, similarly, complex mammals like whales use body gestures like splashing their tail or making bubbles and singing songs to communicate their frustration or affection to other whales and sea creatures.

Human emotions are a complex phenomenon which are a combination of psychological and physiological factors. Emotions are expressed in humans in three different ways, which are:

- subjective experience (conscious experience)
- physiological changes in body (tense muscles, dry mouth, sweating, etc.)
- behaviour of individual (facial expression, flee, hide, etc.)

Emotional speech is a by-product of all of these processes. Emotions are imposed onto the speech when a person experiences some conscious disturbance. This produces physiological changes in the body which results in the production of speech that carries the emotional state of the speaker. Detecting this state just by speech is not very easy even for human beings who, with all their intelligence, still make many mistakes.

One important aspect of emotions is that according to Darwin's theory of evolution, living in the same community with similar group of people does create a particular style of speaking which is generally common to all people of the society. However, each individual is different and has his/her own way of speaking and portraying emotions. There are several factors that affect this style. The style varies from each gender (male or female), age (child,adult or mature), language, culture, and social status of speakers. As we grow in age, our personality develops, we develop our attitude and our methods of interacting and representing ourselves become well defined. As we go through these changes, our method of portraying emotions also changes. Therefore, the way a 10 year old child portrays his/her emotions will differ from how a 40 years old portray their emotions.

As an example of cultural differences, there are several Asian cultures in which full blown expression of *angry* emotions is not very much accepted in the society. Usually people refrain from expressing full blown *anger* which is considered as a weakness of personal control. Similarly, the style of expressing emotions in certain social classes varies from other social classes. The emotions expressed by a worker at the docks will be different from those expressed by a professor in a University. These differences make the task of emotion recognition even more difficult.

Emotions are usually defined as how other people perceive them. This is why we use human listeners to label the recorded speech subjectively in the listening tests i.e., according to how they perceive the corresponding emotions and not the way speaker intended to portray them. The early work on developing basic theories for emotion processing was done by psychologists on the basis of human perception. These basic theories were developed on the full-blown emotions for some discrete number of emotion types mainly portrayed by actors, see for example Murray and Arnott (1993); Scherer (2000).

In a broader sense, we can say that the way human portray emotions should not be significantly different from the way we perceive them. However, up till now, we have not come across any study which tries to link the two or establishes their relationship. As these theories have been developed using human perceptions models, they may not always fit exactly on the models derived directly from the audio speech data.

The most commonly used model for human emotion perception is the dimensional 'valence–arousal' model. All emotions are thought to be the combination of these two dimensions. It is based upon how human listeners perceive emotions rather than how human speakers portray the particular emotions. These methods are good for basic work in the field but the models generated by this type of data do not necessarily fit the real world data. Increasingly, the evidence shows that recognising emotions in *arousal* dimensions is much easier than those in *valence*. This is the prime example of a theory developed by psychologists for which the experimental data does not completely agree.

Another major issue in speech emotion recognition is the identification of the base unit that should be used for recognising emotions. Some researchers advocate using the whole utterance or turn for recognising emotions, while others argue that there could be more than one perceived emotion in the spoken utterance. Therefore, a base unit smaller than the whole utterance or turn should be used. Based upon these different views, databases have been created which are either annotated on the whole utterance level (Engberg and Hansen (1996); Burkhardt *et al.* (2005); Jovicic *et al.* (2004)) or on word level (Batliner *et al.* (2004); Steidl (2009)). General consensus is towards using whole utterance for acted emotional speech while each word for detecting emotions from spontaneous speech. However there are still researchers who use a different unit for this task. For example, FAU Aibo emotional corpus, also used in this thesis, have been labeled on 'chunks' which are predefined segments of speech by the creators.

Another problem is the identification of the features that work well under different emotion conditions. As there has not been much collaboration between researchers from the beginning, there exists many different types of hand crafted base features that have been found to be useful in their specific scenarios e.g., Oudeyer (2003), Fernandez and Picard (2005), Lee and Narayanan (2005) each have used a different feature sets for the task of emotion recognition from speech. This situation seems to be changing by the introduction of OpenEAR toolkit by Eyben *et al.* (2009). This toolkit allows researchers to extract a large number of emotion related features from the speech sample by brute force. This allows to set a base line method for extracting features. Afterwards one can apply some feature selection for the reduction of dimensionality.

Most of the commonly available databases for emotion recognition research are in different languages. Apart from creating problems of language, speaker and recording environment differences, it presents an opportunity to test a very important question *Are emotions universal across different languages?* This means by training on one or more databases and then testing on another which is in a different language, can we learn something about the underlying emotions even though we have no information about the language being spoken. There is evidence that young children can detect emotions and respond to them even though that they do not understand the language.

It is believed that human beings can recognise emotions from speech even if it is expressed in a foreign language that we do not understand. Similarly machines should be capable of understanding emotions regardless of the language and linguistics. Based upon the results on several languages in Russell *et al.* (1989), the authors have argued that valence-arousal model can be considered as a universal model for emotion perception across various languages. If we wish to test the presence of this universal model and correspondingly emotions across several languages from audio speech using machine learning algorithms, we have to establish a universal acoustic feature set that is independent of the language. This will allow us to apply multi-lingual emotion recognition and several emotional speech theories on different languages. Similarly, there have been some initial

experiments performed in which listeners were asked to identify emotions from foreign languages (e.g. Tickle (2000); Abelin and Allwood (2000)). These experiments did find the presence of universality of emotions over various languages. However, there are not many detailed experiments or methods been developed to verify this claim.

The question of the best performing set of features still remains unanswered. Selected features which perform well on one database may not perform as well on another database. Usually, we can not apply exhaustive search for the best performing feature set in a reasonable time, so we have to apply suboptimal and computationally inexpensive feature selection methods. The problem is exaggerated by the fact that best performing features selected by one feature selection method are not same to the ones selected by another even on the same database. These leads to another related question, *Can we search for some universal features which perform equally well on all of the databases?* If we can identify a set of universal features which is common across all/most of the languages, it will allow us to perform cross lingual emotion recognition and answer the questions about the universality of emotions.

Other than learning universal features, another way to test universality of emotions is to learn from the available resources and then apply them on the unknown, not seen before but related tasks. One can take an example of the recently introduced personal assistant in iPhone called Siri by Apple Incorporation. This functionality is only available in English, French and German. The expected date for its release for other languages is more than one year later than the first release of the product. Could there be methods that can be used to learn from the available data and adapt the learned models for the new and unknown languages?

The researchers have started to combine several different classifiers to create an ensemble classifier for emotion recognition. Schuller *et al.* (2005a) used an ensemble of support vector machines, naive Bayes, C4.5 and k-NN classifier on Berlin emotional speech database. However, the trends changed after the first Interspeech Emotion Challenge held in 2009. Most of the contributors in this challenge used binary classifiers arranged in a serial, parallel or a combination of both to form hierarchical classifier. These classifiers have proved to perform better than the others. Lee *et al.* (2009) was the submission which was declared as winner in the classifier sub-challenge. They used binary Bayes logistic regression classifiers to make a hierarchical structure which was based upon their 'prior experience'. There was no formal method described to determine the hierarchical structure of the classifiers.

Most of the research in this domain does not consider the cultural and gender effect on the expressed emotions. Generally, there are no formal methods developed or applied to reduce the differences caused by inter-speaker or gender variations and acoustics of recording environments. Any gender independent SER should explicitly compensate for

these differences. This area of research has been generally overlooked by the researchers. In this thesis we shall try to tackle some of these issues and propose possible solutions.

1.2 Aims of the Thesis

Keeping in view the current issues in this domain, the following are the aims of this thesis:

- to develop effective algorithms for classification of human emotions from their speech
- to identify different characteristics of human speech by applying novel feature selection methods, that represent the emotional state of the speaker
- to address the issue of a universal feature set for emotion recognition that works generally well on all databases
- to get insight into the commonly used 'valence–arousal' model by testing on these two dimensions
- to test methods from different areas of research to improve the generalisation capabilities of SER systems
- to test whether emotions are universal across different languages

1.3 Applications of Work

Call centres are one of the most popular applications of automated recognition of emotions from speech. On one hand, SER systems can provide human operators with information regarding the emotions that their voice might portray. This kind of system helps the users to improve their interaction skills. An example of such a system is Jerk-O-Meter available at http://groupmedia.media.mit.edu/jk.php (last visited on 2 June 2012), which is a real-time speech analysis application that monitors the activity of the user from speech, and gives feedback if they are 'being a jerk' on the phone. On the other hand, such a system may use knowledge of the user's emotional state to select appropriate conciliation strategies and to decide whether or not to transfer the call to a human agent. An example of such a system is AT&T's spoken dialogue system known as 'How may I help you?' in which an automated system responds to the callers and based upon their frustration level decides whether to transfer their call to a human agent or not. The same solution can be used to monitor the quality of the service providers.

Recently, emotion recognition methods have found their applications in computer enhanced learning methods. The motivation behind these methods is to adopt the automated tutoring system according to the emotional state of the student to enhance the learning process. For example, Ai *et al.* (2006) used the features extracted from the dialogue between the automated tutor and the student for emotion recognition in ITSpoke, which is an intelligent tutoring spoken dialogue system. The system adapts its tutoring style based upon the interest level of the student.

Hopkins *et al.* (2011) have used software called *FaceSay* for teaching social skills to children affected with Autism Spectrum Disorders (ASC). A total of 49 children with different levels of ASC were asked to interact with FaceSay routinely to teach them to recognise emotions and how to respond accordingly. An avatar mimicked several emotions and students were rewarded if they recognised and responded to corresponding emotions correctly. This study showed a significant improvement in the capability of students playing this game.

There has been recent work on emotionally aware in-car systems. This work is motivated by studies that provide evidence of dependencies between a driver's performance and his or her emotional state (Jones and Jonsson, 2008). Emotion recognition from speech in the noisy conditions of a moving car has been investigated in the FERMUS project (Schuller *et al.*, 2007a). The aim is to categorise the current state of the speaker to improve and monitor in-car safety and the performance of the driver.

One of the important applications for advancement in this field is the use of emotion recognition in human computer interaction. Researchers are working to develop robots that can interact with humans and modify their actions according to feedback from human reactions. The Sony robot is one example in which a robot called artificial intelligent robot (AIBO) has been developed which can interact with human beings. SER systems have a high potential in games (Jones and Jonsson, 2008), e.g., Microsoft XBox 360 released a concept game character called *Natal* which can monitor and react precisely to the player's style of speech. It can understand and behave accordingly to the emotional response of the player. Another commercial application of this research is the marketing industry. The Affective Computing Group at MIT has developed wrist bands which can monitor the physiological changes which correspond to the emotional state of the user. This information is potentially very useful for the retail industry which can identify the type of products that please or displease their customers.

All in all, SER systems have a large application potential for emotion aware speech recognition systems. However, there are many technical issues that need to be solved before these systems will become part of our daily life technology.

1.4 Thesis Contributions

Contributions of this thesis can be broadly divided into following three parts.

Feature Selection Methods

In this thesis, instead of using limited number of hand crafted features, we have used brute force method to generate a large number of acoustic features. To select the best performing features on an individual database, we have tested several wrapper and filter based feature selection methods. In our experiments, we found that wrapper methods perform better than filter based feature ranking methods but they are computationally expensive. Filter based methods are computationally inexpensive but they do not perform consistently. Two filter based feature selection methods may select different top performing features even when tested on the same database.

We propose a novel feature ranking method, based upon preferential Borda voting scheme. This method fuses the results of several computationally inexpensive filter feature ranking methods using soft and hard decision Borda voting. By fusing several ranked features, we achieve two goals; one is that we get rid of any random ranker, and second is that we find the overall best ranked features.

After selecting features that work well on individual databases, the next step is to search for a universal feature set. We propose to use Borda voting to select a universal feature set that performs reasonably well on all of the selected databases. The proposed methods return two universal feature sets. The classification results using these universal feature sets show the effectiveness of these features. We have used these universal features for inter–database emotion recognition tests where they prove to be very effective.

3DEC Hierarchical Classifier

Using the selected features, we compare four ways to extend binary support vector machines (SVMs) to multiclass classification for recognising emotions from speech—namely two standard SVM schemes (one-versus-one and one-versus-rest) and two other methods (DAG and UDT) that form a hierarchy of classifiers. Analysis of the errors made by these classifiers led us to apply non-metric multi-dimensional scaling (NMDS) to produce a compact (two-dimensional) representation of the data suitable for guiding the choice of decision hierarchy. The distribution of emotion classes on these NMDS representation can be interpreted in terms of the well-known valence-arousal model of emotion. We find that this model does not give a particularly good fit to the data: although the *arousal* dimension can be identified easily, *valence* is not well represented in the transformed data. This is one of the reasons that emotions in this dimension are difficult to classify. We describe a new hierarchical classification technique whose structure is based on NMDS,

which we call Data-Driven Dimensional Emotion Classification (3DEC). This new method performs significantly better than the other methods compared. One can use this systematic data driven approach to guide the hierarchical structure of the classifier.

Importance Weighted Inter–Database Classification

Each speaker has a different style of speaking. These differences are more pronounced between the male and female speakers. Secondly, different and changing acoustic environments are another source of inconsistencies in the data. Any gender and speaker independent SER must be able to generalise well for these varied conditions. This area of research have been generally ignored by the emotion recognition community. We test several algorithms that can be used to enhance the generalisation capabilities of a SER system across varied speakers and acoustic environments. We use standard algorithms like cepstral mean normalisation and maximum likelihood linear regression, which are used in automatic speech recognition systems to compensate for these differences. We identify these differences due to different speakers and recording environments as a covariate shift between the training and testing data. We then propose to use algorithms from the emerging field of *transfer learning* to model this shift by calculating the importance weights (IW) from input data. These IW are used to shift the decision boundary of the classifier towards those training samples which give better representation of the test data.

We have derived mathematical formulation to incorporate these IW into support vector machines (SVMs) which is one of the most commonly used machine learning algorithms. These importance weighted support vector machines (IW-SVMs) are used to cater for the covariate shift in the data. Our experiments on intra–database emotion classification show that transfer learning can be successfully used to compensate for the different speakers and acoustic environments even better than the standard methods used in ASR systems.

To test the real life scenario for a standard SER system, we apply inter–database emotion classification. These tests guarantee that the speakers and environments are different between the training and testing datasets as no part of them are common. To transfer the knowledge gained on the available resources to unknown datasets, we again propose to use transfer learning. Our comparative results show that these algorithms improve the generalisation capabilities of the classifier significantly across the databases which have different speakers and recording acoustic environments.

To answer the question about the universality of emotions across different languages, we have applied inter–database emotion classification using transfer learning on the databases which are in different languages. The results are very successful and we have been able to achieve much better accuracy than the chance

level proving that *there are universal aspects of emotions that are common among different languages.*

1.5 Thesis Organisation

In Chapter 2, we discuss background information about speech production in human beings and the source-filter model used to describe this production system. The basic models used by researchers for describing the emotional states and their interrelationships are also discussed. A general review of current literature on SER systems is also given in this chapter. However, we have reviewed the related literature in every corresponding chapter separately.

Chapter 3 gives the details of the machine learning algorithms used for classification tests. We have also given the details of the two validation techniques used to get the generalisation capabilities of the classifier. In the end we have given the differences between several evaluation measures being used to report classification results along with their mathematical details. This chapter also gives the introduction to the three acted and two spontaneous emotional speech databases which have been used to test the methods in this thesis.

Chapter 4 gives the details of different types of feature sets currently being used for emotion recognition from speech. It gives the details of the large feature set generated by brute force that we use for emotion recognition.

Chapter 5 gives the details of several wrapper and filter based feature selection methods tested in this thesis. It also gives the details of our proposed preferential voting based feature ranking method. Comparative results show the superior capabilities of our proposed method. In the end, we propose a unique way for searching 'a universal' feature set for emotion recognition from speech that works well for all of the databases.

Chapter 6 gives the comparison of four ways to extend binary SVMs to multiclass classification. This chapter also gives the details of our proposed 3DEC hierarchical classifier for which the input data determines the structure of hierarchical classifier.

Chapter 7 gives the details of several algorithms used in ASR systems and the domain of transfer learning, that can be used to enhance the generalisation capabilities of SER system. This is the first time that any such methods have been applied to SER systems. In this chapter, we also test on inter–database emotion classification by training on one database and testing on another unseen database. The question of 'universality of emotions across different languages' is also tackled in this chapter.

We give the future directions for research of this work and conclude this thesis in Chapter 8.

1.6 Publications

The work in this thesis has contributed in part to the following publications:

- A. Hassan and R. I. Damper (2009) 'Emotion recognition from speech using extended feature selection and a simple classifier'. In Interspeech 2009, Brighton, UK. pp. 2403-2406.
- A. Hassan and R. I. Damper (2010) 'Multi-class and hierarchical SVMs for emotion recognition". In Interspeech 2010, Makuhari, Japan. pp. 2354-2357.
- A. Hassan and R. I. Damper (2012) 'Classification of Emotional Speech using 3DEC Hierarchical Classifier' In Journal of Speech Communication. (Accepted for publication Feb 2012)
- A. Hassan, R. I. Damper and M. Niranjan 'On Acoustic Emotion Recognition: Compensating for Covariate Shift' (In preparation)
- A. Hassan, R. I. Damper and M. Niranjan 'On Acoustic Emotion Recognition: Compensating for Covariate Shift – Supplementary Results' Technical report (2012), Published by Faculty of Physics and Applied Sciences, University of Southampton. Available at http://eprints.soton.ac.uk/337383/

Chapter 2

Background and Literature Review

In this chapter, we discuss background information about the production of speech in human beings and the source-filter model used to describe this production system. Emotion production in human beings is discussed along with the details of the theoretical models used for emotion taxonomy. We give a thorough literature review of the past work as well as the current trends in speech emotion recognition systems.

2.1 Description of Speech

The speech production system in human beings is capable of producing highly complex and variable sounds. A cross-section of human speech production system is shown in Figure 2.1 which shows its different parts. All of these parts work in conjunction with each other to produce speech which is radiated from the lips and nose. The quantity and quality of speech produced depends upon the the nature of excitation signal coming from the lungs and the shape and length of the vocal tract which modulates it to produce the final speech. The human vocal system can be divided into two sections: the lungs and the glottis which generate the source signal and the vocal tract which shapes the signal acoustically to produce different types of speech. The glottis is the combination of the vocal folds and the space in between the folds.

2.1.1 The Lungs and Vocal Folds

The power needed to generate the sound comes from the lungs via the breathing mechanism. Mainly exhaled air is used for speaking and the respiratory system can be controlled by the brain so that breathing fits the type of speech being produced. Speech

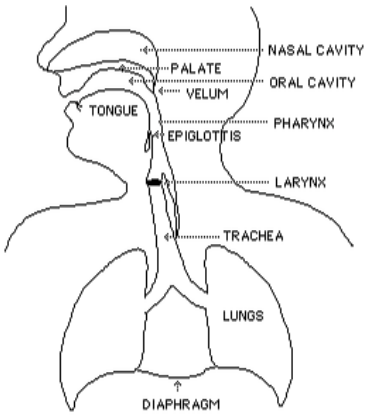


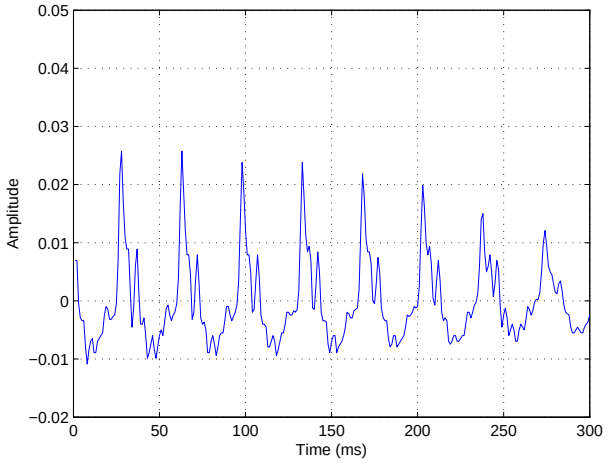
Figure 2.1: The human speech production system, taken from Google Images.

can be voiced or unvoiced. For voiced speech, the vocal folds are brought together, temporarily blocking the flow of air from the lungs and leading to increased pressure. When this pressure becomes greater than the resistance offered by the vocal folds, they open letting the air out. The folds then close rapidly due to a combination of factors, including their elasticity, muscle tension, and the Bernoulli effect. If this process is maintained by a steady supply of pressurised air, the vocal folds will continue to open and close in a pseudo-periodic fashion. As they open and close, pulses of air flow through the glottal opening into the vocal tract. These regular pulses form the excitation signal. The frequency of these pulses determines the fundamental frequency (f_0) which is responsible for the perceived pitch of the produced sound. This is the characteristic of voiced speech. During the process of opening and closing of the vocal folds, if steady pressure is not maintained, a random noise-like pattern is generated which is used to produce hiss-like sounds, the characteristic of unvoiced speech.

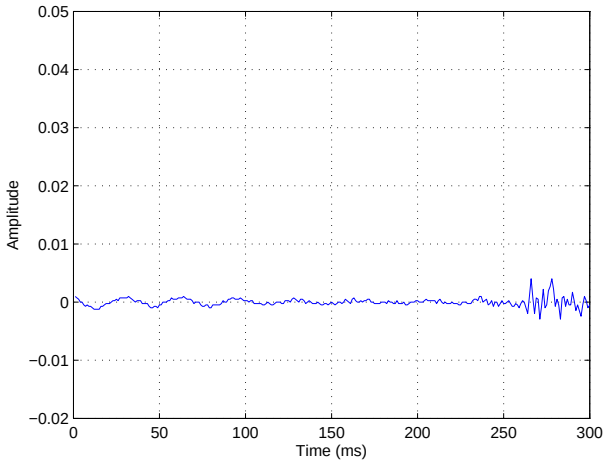
Figure 2.2 shows voiced and unvoiced speech signals. Periodicity of the speech signal due to periodic excitation for voiced speech is clearly visible in Figure 2.2(a) while a random pattern of excitation for unvoiced speech is seen in Figure 2.2(b).

2.1.2 The Vocal Tract

The vocal tract is a passage that connects the larynx to the pharynx, through the mouth and nose to the outside world as shown in Figure 2.1. The section of vocal tract, which lets air out into the atmosphere, consists of mouth, tongue, jaws and lips to form the oral cavity. When the soft velum, see Figure 2.1, is lowered, the oral cavity gets connected to the nasal cavity. The second passage, between the larynx, nostrils and outside air, is called the nasal cavity. The vocal tract acts as an acoustically resonant tube in which



(a) Voiced speech signal.



(b) Un-voiced speech signal.

Figure 2.2: Two speech signals showing (a) voiced speech with clear periodic excitation; (b) unvoiced speech showing random pattern of excitation.

the oral and nasal cavities work together as a musical instrument to produce sound. Waves are reflected from the walls and mouth, and undergo interference which causes some frequencies to be suppressed while others are enhanced. The frequencies at which the vocal tract gives high resonance response are called formant frequencies. The centre frequency and bandwidth of formants are determined by the overall shape, length and volume of the vocal tract. Different sounds are produced by different shapes of the vocal tract. As its shape changes, so does the resonance response, which results in the production of different sounds.

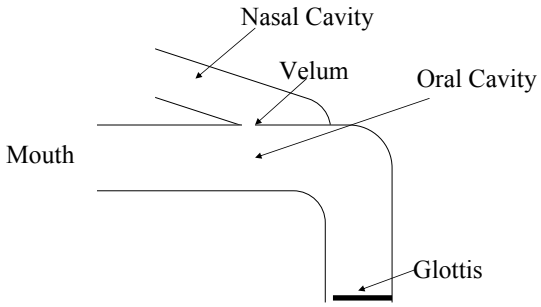


Figure 2.3: A simplified view of human vocal tract as a combination of two tubes.

2.2 Human Speech Production Model

Regardless of different languages spoken, all human beings use the same basic mechanism to produce speech. By careful observation, researchers have generalised this into a simple model for studying its details and understanding and reproducing the human speech.

To generalise the production of speech in human beings, the vocal tract is modelled as a combination of two tubes: the nasal and oral cavity as shown in Figure 2.3. Together they act as an acoustic resonant tube which filters out some of the frequencies while enhancing others. Formant frequencies and their bandwidths are the main characteristics of the oral cavity. The resonant tube can be modelled as an all-pole filter in which the locations of the poles determine the formant frequencies. At the other end of the tube is the source of speech production that will generate the pulses of waves at regular pitch intervals for voiced sounds and random noise-like wave pattern for unvoiced sounds. Pitch and energy are the main characteristics of the source while formants are the characteristic of vocal tract. The idea of air from the lungs as a source and vocal tract as a filter that shapes the sound is called the *source-filter* model for speech production (Fant, 1960) and is regularly used in speech recognition, synthesis and coding applications.

Speech plays the most important part in the process of communication between human beings. Quast (2002) argues that a speech signal carries multiple communication contents: verbal and non-verbal. The verbal contents are the language and words that represent 'what' is being said while the non-verbal contents contains the information of 'how' the words are being said. Every language has some particular words to express a specific emotional state, e.g., if a person is in a happy state, in English the speaker might use words such as *great*, *fantastic*, *brilliant*, etc. Such linguistic features are a rich source of information for humans to determine the current state of the speaker. Even when the verbal contents remain the same, non-verbal channels can carry different information. They convey the intentions of the message to the listeners. As an example "Yes, sure" is a positive sentence in which the speaker is agreeing for something while "*Yes*, sure" is an ironic statement. Although the verbal contents in the sentence have remained the

same, the non-verbal contents have changed. This change conveys the current emotional state of the speaker which is the main topic of interest of this thesis.

2.3 Emotion Production in Human Beings

The question about the nature of the emotions and their affect on speech is an old one. Some theories, like discrete emotion theory and 'big six' emotions, can be linked back to Darwin. Before we delve into their details, it is important to give an acceptable definition and the characteristics of 'emotion'.

In his paper, Scherer (2000) has differentiated the emotions from other similar human characteristics like mood and personal attitude. Emotions are usually observed in brief episodes in response to internal or external events of significance. A person's 'mood' is a state of low intensity and can last for hours or days. It can be characterised as cheerful, gloomy, irritable, buoyant, etc. An individual's personal 'attitude' is part of their personality and is unlikely to alter.

In this thesis we follow the following generally accepted definition by Scherer (2000) who defines emotions as:

> "*episodes of coordinated changes in several components in response to external and internal events of major significance to the organism*"

The external events could be a change in the current situation, response of another person etc., while the internal events could be thoughts, memories and sensations. From this definition we can say the following about 'emotions':

- Emotions are episodic in nature, they occur again and again depending upon the stimuli
- The effect of emotions on a person is noticeable and measurable
- The effect of emotions last for some measurable time
- The external and/or internal stimuli are very important to the generation of emotions
- Emotions are generated in human beings as a result of complex neuro-physiological processes, which affect several traits of normal human behaviour may it be verbal, physiological or facial behaviour.

According to psychological studies into the mechanism of emotion production, it has been found that under the influence of joy, anger and fear stimuli, the human nervous system is aroused. This causes an increase in heart rate, blood pressure, breathing patterns,

dryness of mouth and occasional muscle tremor. The resulting speech is correspondingly loud, fast and has high pitch range. In a contrasting situation like sadness, heart rate and blood pressure decreases and salivation increases, producing speech that is slow, low pitched, and low in energy. All of these changes in the human physiology affect the produced speech, facial expressions and human response to the stimuli like fleeing in case of fear. These physiological changes are also used in the polygraph test to check if a person is telling the truth or not.

2.4 Theoretical Models for Emotion Taxonomy

An important issue in developing a speech emotion recognition (SER) system is the need to determine and identify a set of emotions that are to be recognised and classified by a machine. In the last couple of decades, all the research related to emotion taxonomy is generally based upon two schools of thoughts. One assumes a discrete number of basic emotions that are representative of all other emotions. This is very similar to the idea of seven colours of light that can be combined to generate any colour. The second school of thought believes in continuous emotions and proposes to map all emotions on a low dimensional space, usually 2-3 dimensions, and argue that all emotions can be generated by combining the continuum of the underlying dimensions.

2.4.1 Discrete Emotion Theory

In the literature, up to 300 emotions have been identified by linguistics like O'Connor and Arnold (1973), however identification and recognition of such a large number of emotions is very difficult especially by machines.

Therefore, based upon the idea of seven discrete colours of light, most researchers use the 'palette theory' (Scherer, 2003) which argues that all of the emotions are a combination of a few basic emotions. The number of basic emotions varies from 6–14. They are characterised by specific response patterns in physiology of human beings as well as their vocal and facial expressions. Many scientists have defined a list of emotions that they consider to include the most basic and important emotions. Table 2.1 is reproduced from Ortony and Turner (1990) which summarises the list of basic emotions used by various researchers.

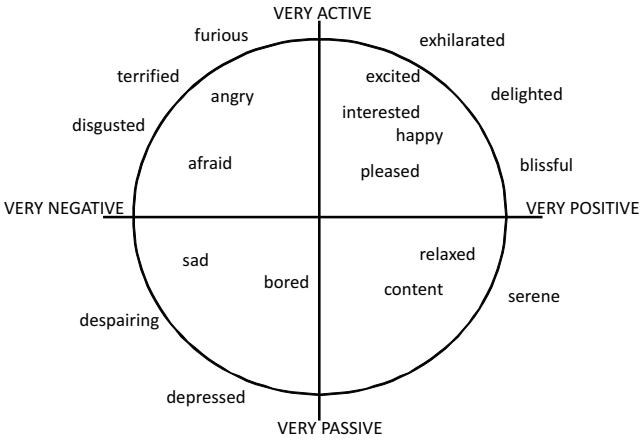
The term 'big six' (Cowie *et al.*, 2001) has become quite common for the description of discrete emotions. It implies the existence of six basic emotions. Most commonly listed emotion are *angry*, *sad*, *surprised*, *happy*, *fear* and *disgust* with *neutral* sometimes added as a seventh emotion. These emotions are also referred to as 'archetypal emotions'.

Table 2.1: Basic emotion classification by different researchers and scientists (Ortony and Turner, 1990).

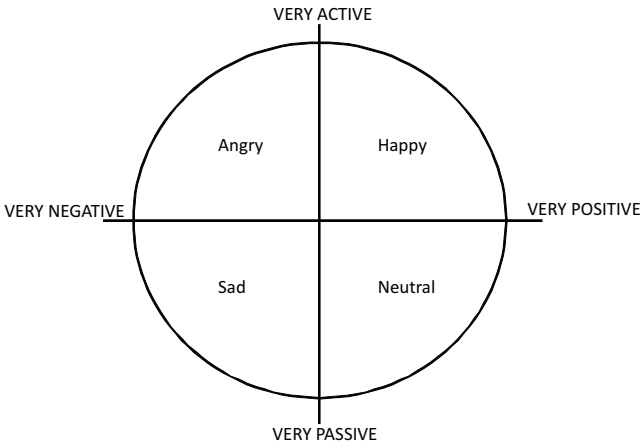
Reference	**Basic Emotion**
McDougall (1926)	Anger, disgust, elation, fear, subjection, tender-emotion, wonder
Watson (1930)	Fear, love, rage
Mowrer (1960)	Pain, pleasure
Izard (1977)	Anger, contempt, disgust, distress, fear, guilt, interest, joy, shame, surprise
Plutchik (1980)	Acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise
Ekman, Friesen, and Ellsworth (1982)	Anger, disgust, fear, joy, sadness, surprise
Panksepp (1982)	Expectancy, fear, rage, panic
Tomkins (1984)	Anger, interest, contempt, disgust, distress, fear, joy, shame, surprise
James (1984)	Fear, grief, love, rage
Weiner and Graham (1984)	Happiness, sadness
Gray (1985)	Rage and terror, anxiety, joy
Frijda (1986)	Desire, happiness, interest, surprise, wonder, sorrow
Oatley and Johnson-Laird (1987)	Anger, disgust, anxiety, happiness, sadness

2.4.2 Dimensional Emotion Theory

In order to understand how human beings conceptualise emotions, the approach used by the psychologists has been to seek the basic dimensions by which we perceive the similarities and differences among various emotions. The idea is to identify the basic dimensions which can define and map all different emotions using a lower dimensional space. This allows the researchers to visualise the complex mechanism of emotion production in human beings in a lower dimensional space. Often these studies on several language have results in two bipolar dimensions, *valence* and *arousal*. In Russell (1980), the author suggested that only two dimensions, 'valence' and 'arousal' are enough to characterise all emotions and proposed the circumplex model of affect for speech. He argued that emotions are spread around the perimeter of valence-arousal space. Arousal refers to the current state of activeness of the speaker while expressing a certain emotion. Therefore, *angry* and *happy* should have high arousal while *sad*, *bored* and *relaxed* have low arousal. The second dimension, 'valence' represents the positive and negative feelings of the speaker. This dimension is used to separate *angry* from *happy* emotions. Sometimes a third dimension namely 'power' or 'control' is added to this model.



(a) Circumplex model of affect



(b) Four primary emotions on the circumplex model of affect

Figure 2.4: Graphical representation of (a) circumplex model of affect with horizontal axis representing 'valence' and vertical axis representing 'arousal'; (b) four primary emotions on circumplex model.

Similar to the spectrum of colour, emotions also lack the discrete borders that would clearly differentiate one emotion from another. Each can be understood as the linear combination of these two dimensions. Figure 2.4(a) shows a number of emotions spread across the two dimensional space. Figure 2.4(b) shows only four emotions *angry*, *happy*, *sad* and *neutral* on the circumplex model. These four emotions are quite extensively used in our studies in this thesis as described in Chapter 7.

Russell *et al.* (1989) have reported on two emotion perception studies; one using the speech from four different languages and the second using the facial expressions from three different languages. The languages used for these experiments were English, Polish, Greek, Estonian, Chinese and Gujrati. In all of their experiments, they identified the valence and arousal dimensions represented in the audio speech as well as in facial representation of emotions. All of these languages are very different from each other

and represent very varied cultures and speaking styles. From their experiments, they conclude that valence-arousal dimensions can be considered as an underlying universal model which is common across many if not all languages.

These traditional dimensions have been developed by psychologists by looking at how a layman perceives prototypical full-blown acted emotions. The situation is slightly different when we deal with real life in which spontaneous emotions are not always full-blown. In such a situation, one can not always identify *valence* and *arousal* dimensions from the speech data. We have discussed this theory in further detail in Chapter 6 in which we have used a data driven approach for identifying the underlying dimensions rather than psychological information.

However, these results must be interpreted with care. The results reported in Chapter 6 are based upon how the several emotions can be mapped onto a lower dimensional space and how we interpret the basic underlying dimensions from the speech data. These dimensions are derived from the spoken speech data and therefore are slightly different from human perception based valence-arousal model.

The limitation of discrete classification of emotions is that we are trying to quantify emotions which is a subjective feeling. It is hard to quantify a subjective feeling as there are no discrete boundaries. However, to allow machines perform this task, somehow we have to define and discretise emotions into certain number of classes. Different emotional models try to quantify them by either using the discrete models or dimensional models. By using discrete models we are limiting ourselves or the machine to a certain number of emotion classes which will be portrayed in the acted speech but might not be present in the real world spontaneous speech data. The dimensional model gives a bit of flexibility that we can use the underlying dimensions to formulate the current emotional state of the speaker which might not be a standard, prototypical emotion and allows to model much more emotion classes.

2.5 A Review of Speech Emotion Recognition Systems

A SER system is a pattern recognition system that can take an unknown input speech sample and, by using information extracted from the sample, predict the current emotional state of the speaker. A typical SER consists of many components, as shown in Figure 2.5. The foremost requirement for any machine learning algorithm is a database to learn from. For this task, sometimes actors are used to portray the specified emotions or spontaneous emotional speech is collected from speakers engaged in a conversation or wizard-of-oz scenario is used in which the participants complete the tasks without the knowledge that their emotionally coloured speech is of interest. To establish the ground truth on the recorded database, human listeners are used to give particular labels to each

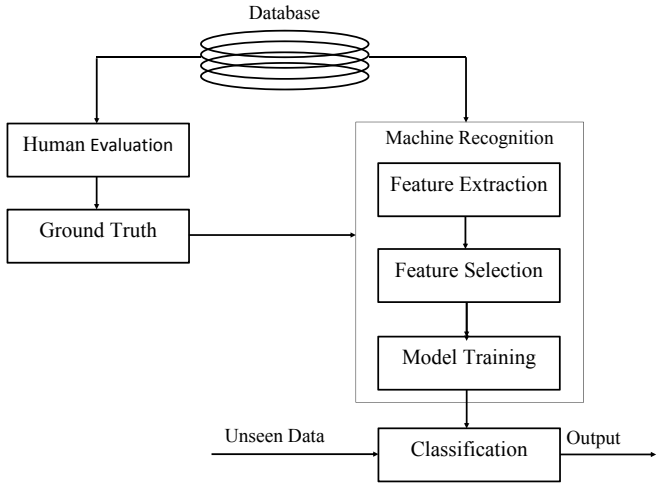


Figure 2.5: Components of a typical speech emotion recognition system.

emotional speech sample. These labelled samples are used to train a machine learning algorithm which is later used for making decisions on unseen test samples.

Many researchers have focused on different components separately and independently. Some have contributed by creating databases which contain emotionally coloured speech that is used by the learning algorithm to model the patterns. Some have worked on the features and types of features that can be extracted from speech which give the best information about the current emotional state of the speaker. Others have worked on the machine learning algorithms and methods that can give insight into the processing of emotions. In the following section, we will look at each component separately by reviewing the relevant literature.

2.5.1 Emotional Speech Databases

From Figure 2.5, it can be seen that first step for a SER system is the availability of emotional speech databases. Douglas-Cowie *et al.* (2003), Ververidis and Kotropoulos (2006) and El Ayadi *et al.* (2011) have listed several speech databases that have been used by many researchers engaged in emotion recognition. Three different types of databases can be observed in the literature: simulated or acted speech in which emotions are expressed deliberately by professionals, natural or spontaneous speech where all recordings are done in real world environment and elicited speech in which emotions are induced by putting the subject into certain controlled conditions. Table 2.2 shows details of the currently used databases.

Dellaert *et al.* (1996) and Murray and Arnott (1993) were among the pioneers of research on emotion synthesis and recognition. They used acted speech, in which emotions are expressed deliberately by professional actors. The advantages of using acted rather than more natural or spontaneous speech are obvious. It is relatively easy to control

Table 2.2: Summary of common emotional speech databases.

Name	Reference	Access	Language	Size	Source	Emotions
Danish emotional speech	Engberg and Hansen (1996)	Public with license	Danish	4 speakers, 260 utter.	Nonprofessional actors	Anger, joy, sadness, surprise, neutral
SUSAS	Hansen and Bou-Ghazale (1997)	Public with license	English	32 speakers, 16,000 utter.	Speech under simulated and actual stress	Stress, fear, anxiety, anger
Interface database	Hozjan *et al.* (2002)	Commercial	English, Slovenian, Spanish, French	42 speakers, English (186 utter.), Slovenian (190 utter.), Spanish (184 utter.), French (175 utter.)	Actors	Anger, disgust, fear, joy, surprise, sadness, neutral
KISMET database	Breazeal and Aryananda (2002)	Private	American English	3 speakers, 1002 utter.	Nonprofessional actors	Approval, attention, prohibition, soothing, neutral
SmartKom	Schiel *et al.* (2002)	Public	English, German	79 speakers, over 3,800 utter.	Wizard-of-Oz scenario	Neutral, joy, anger, helplessness, pondering and surprise
BabyEars	Slaney and McRoberts (2003)	Private	English	12 speakers, 509 utter.	Mothers and fathers	Approval, attention, prohibition
Berlin emotional speech databases	Burkhardt *et al.* (2005)	Public and free	German	10 speakers, over 500 utter.	Professional actors	Anger, joy, sadness, fear, disgust, boredom, neutral
Sensitive artificial listener	Douglas-Cowie *et al.* (2007)	Public	English	125 speakers, 1696 utter.	Audio-visual interaction recordings	Continuous labels on valence and arousal domain
FAU Aibo emotion corpus	Steidl (2009)	Public with license fee	German	51 speakers, 18,216 utter.	Children speech	Angry, emphatic, neutral, positive and rest
Audio-visual interest corpus	Schuller *et al.* (2009a)	Public with license fee	English	21 speakers, over 3,000 utter.	Adult speaker engaged in conversation	Boredom, neutral, joyful
SAVEE database	Haq and Jackson (2010)	Public and free	British English	4 speakers, 480 utter.	Nonprofessional actors	Anger, disgust, fear, happiness, sadness, surprise, neutral
SEMAINE database	McKeown *et al.* (2010)	Public and free	English	20 speakers, 50,350 words	Conversation with human operator	Anger, disgust, amusement, happiness, sadness, contempt

the conditions and serious ethical issues, such as eliciting genuine fear in human subjects or recording sensitive real-world emotion-charged interactions, are avoided. The disadvantages are equally obvious in that acted emotional speech is not fully natural, such that strong objections have emerged recently against its use (Vogt and André, 2005; Wilting *et al.*, 2006; Shahid *et al.*, 2008; Schuller *et al.*, 2009b). Wilting *et al.* (2006) and Shahid *et al.* (2008) have given further insight into the differences between acted and spontaneous emotions by showing that the acted emotions are different from natural/spontaneous emotions in human perception tests.

For acted emotional speech, usually trained actors are used which belong to educated cosmopolitan middle class society. Their method of portraying emotions will be different from a person who works at a tea shop. Due to these differences, the models calculated from actors can not be directly used on the data collected from speakers belonging to other levels of social hierarchy. This is one of the drawbacks which is raised against using acted emotional speech for emotion research that it only depicts a certain people belonging to educated, middle class society and does not cover the whole spectrum of different people belonging to different social status of the society.

Spontaneous emotions may be shaded, weak and hard to distinguish as compared to acted emotions which, being deliberate, are in most cases somewhat overdone and well-differentiated. Therefore, the complexity of the task increases as we move from acted to spontaneous speech. Nonetheless, the ease with which acted speech can be controlled, together with the difficulties of collecting natural emotional speech under realistic conditions, means that much current research still continues to use acted data.

The acted emotional speech databases are obtained by asking actors to speak some sentences with a predefined emotion in a controlled environment. The most popular databases are the Danish emotional speech database (DES) (Engberg and Hansen, 1996) and the Berlin emotional speech database (Burkhardt *et al.*, 2005) which are publicly available for research use. Both of these databases contain recording for 5 and 7 emotions respectively, out of which four are common among the two. The Serbian emotional speech database (Jovicic *et al.*, 2004) is another database containing acted speech which has been recently compiled for the Serbian language. It contains around 2790 utterances for five different emotions. Another database, although not commonly used, is the Interface database (Hozjan *et al.*, 2002). It is a multilingual database containing seven different emotions in French, Spanish, Slovenian and English. The database was designed for the general study of emotional speech. However, it can be very useful for inter-lingual emotion testing. Nogueiras *et al.* (2001) used it for emotion recognition in Spanish only while Sidorova (2009) has used this database for inter-lingual emotion detection.

One way to obtain spontaneous speech data is to record speakers during the interaction with other agents, where the interaction is likely to elicit certain emotions but is unlikely that these will be too extreme (e.g., leading to threats of violence). For example, Kismet

(Breazeal and Aryananda, 2002) is a database of infant- and robot-directed speech, and BabyEars (Slaney and McRoberts, 2003) contains recordings of parents talking to their children and expressing different emotions like approval, attention, and prohibition. The speech under simulated and actual stress (SUSAS) database collected by Hansen and Bou-Ghazale (1997) contains isolated-word utterances produced by 32 speakers under conditions of stress and emotion, including both acted and spontaneous speech. The latter is obtained, e.g., during roller coaster rides.

Some researchers have collected speech in more naturalistic interaction situations. For example, Lee and Narayanan (2005) recorded calls in a 'live', commercially-deployed call centre setting, in which customer frustration was expected to give rise to negative emotions. Other people using similar ideas are Balentine and Morgan (2002), McTear (2002) and Vidrascu and Devillers (2005). Working with this type of data is especially difficult as spontaneous emotions are hard to distinguish and the proportion of data displaying particular emotions can be sparse with respect to the whole database. Another consideration is that these databases are not usually publicly available because of commercial and copyright issues. As well as limiting the possibility for researchers to use them, this also renders the results obtained problematic because they cannot easily be replicated and validated by others.

Yet another possibility is to provoke or induce emotions in speakers by putting them into certain controlled conditions without them knowing that their emotional state is of interest. Examples of databases collected in this way are SmartKom (Schiel *et al.*, 2002) and the German FAU Aibo Emotion Corpus (Batliner *et al.*, 2004), hereafter "Aibo". Here, the creators have used Wizard-of-Oz techniques to induce emotions in a controlled environment by asking the subjects to complete certain tasks that were manipulated and controlled by the wizard. Strictly speaking, these emotions are induced and this type of database should be called 'elicited'. However, this type of data is the closest that we can get to spontaneous speech without encountering ethical or commercial issues and as current literature refers to it as 'spontaneous', we will follow this general convention here.

In recent years, some audio-visual databases have been recorded. The Belfast sensitive artificial listener (SAL) (Douglas-Cowie *et al.*, 2007) is an audio-visual emotional speech database which contains clips from television shows and interviews. The data is labelled on the continuous valence and arousal scale. Other similar databases are the audio-visual interest corpus (AVIC) (Schuller *et al.*, 2009a) and the Surrey audio-visual expressed emotion (SAVEE) database (Haq and Jackson, 2010). The SEMAINE database (McKeown *et al.*, 2010) has been the integral part of the recently held Audio/Visual Emotion Challenge 2011 (Schuller *et al.*, 2011a). All of these have been collected to test the effectiveness of audio and visual channels both together and independently.

Issues With Existing Emotional Speech Databases

There are several issues that need to be addressed to advance the research in the area of emotion recognition. Some of the most important issues are listed below:

- Most of the databases available for public research contain acted speech and are small in size (approximately 500–1000) sentences spoken by a small number of speakers, approximately 5–10.
- There are not many large spontaneous speech databases freely available to be used by SER systems.
- Due to the nature of the spontaneous data, the target classes are not usually balanced. Hence one has to apply sample balancing to train a classifier on such an imbalanced datasets.
- As each database is recorded for a specific purpose, the recording environments are different for each situation. No effort has been made to standardise the recording equipment and environments across the databases.
- The base unit used for annotation differ between several databases, e.g., SmartKom is annotated on the frame level, FAU Aibo on the word level, valence and arousal dimensions in SAL, DES and Berlin on the utterance level.
- There is no standard way to choose the quality and quantity of labellers. As an example, for FAU Aibo emotional speech database, 5 experts labelled the data, while for Danish emotional speech database, 20 students did the labelling. The number and methods for choosing labellers is different for each database.
- Because of different annotations, the number of emotion categories per database is also different. As these are inconsistent across the databases, inter-database testing is not straight forward.
- Many databases are in different languages and it is not easily possible to use language contents of one database for making decision on another language.
- Usually phonetic transcriptions are not provided with most of the databases because of which it is difficult to extract linguistic contents from the utterances of such databases.

2.5.2 Features for Speech Emotion Recognition

An important issue in the design of an SER system is the extraction of suitable features from speech samples that can efficiently and correctly characterise emotions. The early and much-cited work of Murray and Arnott (1993)—since verified and updated (Murray

and Arnott, 2008)—identified pitch, intensity, speaking rate and voice quality as the acoustic features most affected by emotions in speech. As recordings were from actors, Murray and Arnott assumed that the emotions would remain constant over the whole utterance and so it was treated as a single unit. This assumption has been accepted as the basis of most subsequent studies. However, researchers who work on spontaneous speech have questioned if this is appropriate. As emotions in spontaneous speech can be short-lived, the features should perhaps be calculated over smaller units than utterances. Some studies (e.g., Shami and Verhelst, 2007; Casale *et al.*, 2008) have tested this by dividing the whole utterance into segments containing only voiced speech; features are then taken from these voiced segments alone. Features calculated from segments are sometimes referred to as *short-term* whereas those obtained from the whole utterance are called *long-term* (Li and Zhao, 1998). Results to date show that attempts to use segment level (short-term) features alone for emotion classification have not been as successful as using utterance level (long-term) features only. However, both Shami and Verhelst (2007) and Casale *et al.* (2008) got better results by combining the two levels, indicating that although long-term features by themselves do best, it is unrealistic to expect emotion to be constant over a whole utterance. In the Interspeech 2009 Emotion Challenge (Schuller *et al.*, 2009b), the FAU Aibo database was divided into manually defined segments of speech called chunks. Similarly, in the 2010 and 2011 Emotion Challenges, the database was divided into words and each word was used for extracting features. There seems to be general consensus that for processing acted emotional speech, utterance level processing is the best while for spontaneous speech, word level processing performs the best.

Different researchers have used many different sets of features for developing their SER systems. Along with the standard prosodic features like pitch, energy, and rhythm, many researchers have found voice quality and spectral features very useful. Oudeyer (2003) used around 200 features related to pitch, intensity and the spectrum of the speech signal and applied genetic algorithm based feature selection. Several machine learning algorithms were applied for recognising four emotional states. In the end an interactive game was developed where a robot was able to recognise the emotions of the user and responded in the appropriate synthesised emotional speech.

Fernandez and Picard (2005) have introduced several voice quality based features and found them to perform very well for emotion recognition. Yang and Lugger (2010) have argued that prosodic features can separate emotion classes in the arousal dimension whereas voice quality features are more effective in separating classes in the valence dimension. Similarly, Eyben *et al.* (2010) and Schuller *et al.* (2011a) have found mel-cepstral features most effective in the valence dimension. Lee and Narayanan (2005) and Yildirim *et al.* (2011) included discourse information to augment the decision of the classifier and reported improved performance of the classifier.

Given this diversity, it is unclear exactly which are the 'best' features to use. Hence, a new idea of 'brute force' approach has emerged. The idea is to pool together a large number of features consisting of different types of feature sets. Batliner *et al.* (2006) were the first to pool together a large number of features from sets independently developed at different sites, yielding a superset of over 6000 features, and found a performance improvement from using such a large number of features. Subsequently, Eyben *et al.* (2009) have described an open-source toolkit called OpenEAR for extracting large numbers of features. This toolkit has become the work horse for the research on emotion recognition.

2.5.3 Classification Methods

The last stage of an SER system is the classifier that makes the final decision about the underlying emotions. Several, traditional classifiers like k-nearest neighbours (k-NN), hidden Markov models (HMM), Gaussian mixture models (GMM), support vector machines (SVM) and multi-layer perceptrons (MLP) have been applied for emotion recognition.

Traditionally, HMMs and GMMs are the classifiers of choice for automatic speech recognition systems. They work well when the frame level information is of interest. As observed in the last section, in emotion recognition long term features usually perform the best, hence SVMs have become the classifier of choice. Kim *et al.* (2007) proposed a very interesting idea of using GMMs for spectral features and k-NN for prosodic features. They combined the results of both classifiers using weighted sum and reported an equal error rate on a database collected at USC, Los Angeles. However, in the reported results, no significant improvement was obtained by using the combined classifiers.

Shami and Verhelst (2007) have used 200 features extracted by Oudeyer (2003) on four databases. They applied four machine learning algorithms on Kismet, BabyEars, Berlin and DES emotional speech database. The first two databases contain infant-directed emotion whereas the last two are acted emotional speech databases. The classification was performed using k-NN, SVM and AdaBoost C4.5 algorithms on several individual databases as well as inter-database classification by using Kismet–BabyEars and Berlin–DES combinations and found SVMs to be performing the best. Inter database classification was performed in three different ways: within database, inter database and integrated database. For inter–database classification, the reported results are just above the chance level. However these results present an interesting insight that something universal can be learnt about emotions by training and testing on similar databases.

Schuller *et al.* (2005b) have collected 2440 speech samples extracted from films containing joy, anger, disgust, fear, sadness, surprise and neutral emotions. They have extracted 276 acoustic features and applied a boosting C4.5 algorithm for classification. According

to the reported results, the boosting C4.5 algorithm was able to outperform SVM based classification.

Schuller *et al.* (2009a) have reported classification results on nine acted and spontaneous speech databases. They used the OpenEAR toolkit to extract 6552 features from each speech sample. A GMM classifier was used to model frame level features and an SVM classifier with a polynomial kernel was used to model long term features. They have reported results on the classification of valence and arousal as well as all of the classes in each database. To reduce the channel effect a normalisation method, known as Cepstral mean normalisation, was applied. From their results it was observed that overall, SVM classifiers performed the best. The authors argue that GMM classifiers performed better for the spontaneous speech databases as compared to acted speech databases. The reason for this improved performance is that in spontaneous speech, emotions vary at a smaller scale than the whole utterance. Hence, a GMM is able to model the changes better in comparison to acted speech where the same emotions are portrayed for the whole utterance.

Eyben *et al.* (2010) have used four databases, SmartKom, FAU Aibo, Sensitive Artificial Listener and Vera-Am-Mittag, for recognition of *positive*, *negative* and *neutral* 'valence' by using 'leave-one-database-out' cross validation. Using the OpenEAR toolkit, 2832 acoustic features were extracted and on average they achieved 53.4% unweighted average accuracy on all four databases using SVM. This accuracy is much higher than chance level (33.3%) on the notoriously difficult dimension of emotions (valence) which is very encouraging.

Schuller *et al.* (2011b) have used six current emotional speech databases for recognising 'valence' and 'arousal' by using the leave-one-database-out method and 6552 features. They have presented results by combining the training data of all the databases as well as combining the results of several classifiers by majority voting. They found that majority voting increased the overall unweighted average (UA) accuracy performance. The overall 64.3% UA for arousal and 56.7% UA for valence was achieved using SVM classifier.

A recent trend is towards using multiple classifier systems for the task of emotion recognition. There are three ways that multiple classifiers can be combined together: serial, parallel and hierarchical. In a serial structure (Planet *et al.*, 2009), classifiers are arranged to form a queue or pipeline, where each classifier reduces the number of classes for the next classifier. In a parallel structure (Yildirim *et al.*, 2011; Lee *et al.*, 2011), several classifiers are used in parallel to make independent decisions and the final decision is obtained by fusing the results together. In a hierarchical structure Yang and Lugger (2010), classifiers are arranged in a tree structure where the number of candidate classes becomes smaller as we go deeper in the tree.

Based upon several studies, we can conclude that the most commonly used classifiers for speech emotion recognition are SVMs. In this thesis we have also used SVM classifiers with features extracted by using OpenEAR toolkit. We have looked at several feature reduction methods and report their results.

2.6 Summary

In this chapter, we have discussed the basic source-filter speech production model to develop its understanding which is helpful in understanding the features that represent the emotional state of the speaker. We have also looked at the theoretical models used for emotion taxonomy and given the details of dimensional emotion model and discrete emotion model. The most relevant model to our work is valence-arousal model. In the second part of this chapter, we have given a thorough literature review of state of the art speech emotion recognition systems.

In the following chapter, we have given the basics of a SVM classifier that we have used in this thesis. The details of the three acted and one spontaneous emotional speech databases along with the evaluation measures used to report our results are also given.

Chapter 3

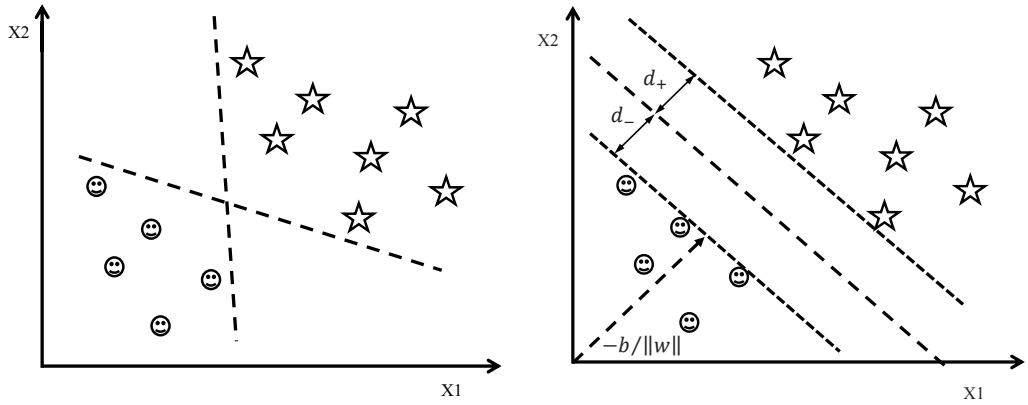
Classification Methods

Any SER system consists of two stages: the front end which acquires the speech signal, does the pre-processing and extracts the appropriate features to be processed by the second stage, a classifier then makes the decision about the underlying emotional state of the speaker in the target utterance. Several traditional machine learning classifiers have been used in SER research and they have been thoroughly reviewed in Chapter 2. The most commonly used classifiers for the task of emotion recognition are hidden Markov models (HMM), Gaussian mixture models (GMM), support vector machines (SVM), multilayer perceptrons (MLP) and k-nearest neighbours (k-NN). Usually, a speech sample is divided into small intervals of 20ms, called frames, to extract features. If the classification is applied on the frame level, then the classifiers of choice are HMMs and GMMs. However, if the decisions are being made on base units larger than a frame, then the classifier of choice are SVMs, which in many studies have proven to be performing the best.

In this chapter, we briefly describe the machine learning algorithms that have been used in this thesis as the basis of the study. We also give the details of the two validation methods that we have used to obtain generalisation performance of the classifier. We then define the performance measure (average accuracy) that is used to report our results and give its equivalence with average recall which has been used for reporting the results of several emotion challenge competitions. In the end we give details of four databases that we have used to report our results.

3.1 Support Vector Machines

Support vector machines (SVMs) are supervised statistical machine learners developed by Vapnik (1995). They can be used for classification as well as regression. They were originally developed for solving linearly separable binary classification problems and were



(a) (b)

Figure 3.1: Linearly separable data in which (a) shows possible hyperplanes that can separate the two classes; (b) shows the optimal hyperplane which separates the two classes.

then extended to non-separable data with the introduction of slack variables. After the introduction of kernels, SVM can be applied to non-linearly separable data. SVMs have been successfully extended to multi-class problems using a combination of several binary classifiers. However, this extension is still an on-going research issue. In Chapter 6 we have tested four methods for extending binary SVMs to multiclass classification.

3.1.1 Linear SVMs

SVMs are based upon finding an optimal hyperplane separating the data of two classes (considering a binary case). To get an understanding of why finding the largest separating hyperplane is a good idea consider the example shown in Figure 3.1(a). It shows a two dimensional linearly separable data of two classes. There are many different ways to select the hyperplane that can separate these two classes without an error as shown in Figure 3.1(a) by dotted lines. However, there is one hyperplane which maximises the margin between the two classes which is shown in Figure 3.1(b). This should give better generalisation of the data than the other two.

An optimal separating hyperplane is one that separates the data of two classes and also maximises the margin of the hyperplane. Consider training data of the form: $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, where each x_i is a D-dimensional vector ($x \in \mathbb{R}^D$), and $y_i \in \{+1, -1\}$ are the corresponding class labels. The separating hyperplane has the form:

$$\langle w, x_i \rangle + b \quad = \quad 0 \tag{3.1}$$

where w is the normal vector to the hyperplane and b is the offset. Without the offset b, the hyperplane has to pass through the origin, restricting the solution. The values of

w and b are learnt from the input data. The input data points are said to be optimally separated by the hyperplane if they are separated without errors, and the distance between the closest point of each class to the hyperplane is maximal.

A point x which lies on the hyperplane satisfies $\langle w, x\rangle + b = 0$ and the perpendicular distance from x to the origin is given by $|-b|/|w|$. Let d_+ (d_-) be the shortest distance from x to the separating hyperplane to the closest positive (negative) example, then the 'margin' of the separating hyperplane is given by $(d_+ + d_-)$.

For a linearly separable case, the SVM tries to find the separating hyperplane with largest margin. This is formulated as: suppose all the training data satisfy the following constraints:

$$\langle w, x_i\rangle + b \geq +1 \quad \text{for} \quad y_i = +1 \tag{3.2}$$

$$\langle w, x_i\rangle + b \leq -1 \quad \text{for} \quad y_i = -1 \tag{3.3}$$

The two equations can be combined into one set of inequalities:

$$y_i(\langle w, x_i\rangle + b) \geq 1, \quad \forall_i \tag{3.4}$$

The distance $d(w, b; x_i)$ of a point x_i from the hyperplane defined by (w, b) is:

$$d(w, b; x_i) = \frac{|\langle w, x_i\rangle + b|}{|w|} \tag{3.5}$$

For a point x_i that lies on the hyperplane $H1 : \langle w, x_i\rangle + b = 1$ with normal w, its perpendicular distance to the origin is $\frac{|1-b|}{|w|}$. Similarly, for a point x_i that lies on the hyperplane $H2 : \langle w, x_i\rangle + b = -1$, its distance to the margin is $\frac{|-1-b|}{|w|}$. Therefore, the margin is equal to $\frac{|1-b|}{|w|} - \frac{|-1-b|}{|w|} = \frac{2}{|w|}$. We therefore look for a hyperplane that gives the maximum margin by minimising $|w|$, using the following form:

$$\min_w \tfrac{1}{2}|w|^2 \tag{3.6}$$

subject to

$$y_i(\langle w, x_i\rangle + b) \geq 1$$
$$1 \leq i \leq n$$

This optimisation problem can be combined into one equation by using Lagrange multipliers as follows:

$$L_P(w, b, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{n} \alpha_i(y_i(\langle w, x_i\rangle + b) - 1) \tag{3.7}$$

where α are the Lagrange multipliers. This Equation 3.7 has to be minimised with respect to w, b and maximised with respect to $\alpha \geq 0$. By taking the partial derivative of the above equation with respect to w and b and putting it to zero we get the following equation:

$$\begin{aligned} L_D(w,b,\alpha) &\equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle - \sum_{i=1}^{n}\alpha_i \quad \text{s.t.} \quad \alpha_i \geq 0, \sum_{i=1}^{n}\alpha_i y_i = 0 \\ &\equiv \frac{1}{2}\sum_{i,j}^{n}\alpha_i H_{i,j}\alpha_j - \sum_{i=1}^{n}\alpha_i \quad \text{where} \quad H_{i,j} \equiv y_i y_j \langle x_i, x_j\rangle \\ &\equiv \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} - \sum_{i=1}^{n}\alpha_i \quad \text{s.t.} \quad \alpha_i \geq 0, \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{3.8}$$

which is a function of α_i only. Equation 3.8 is called the dual of Equation 3.7 and its solution can be found by only minimising with respect to α_i. The parameters of the optimal hyperplane are then calculated as:

$$\begin{aligned} w^* &= \sum_{i=1}^{n}\alpha_i y_i x_i \\ b^* &= -\frac{1}{2}\langle w^*, x_r + x_s\rangle \end{aligned}$$

where x_r and x_s are any support vector from each class satisfying

$$\alpha_r, \alpha_s > 0, \quad y_r = 1, y_s = 1 \tag{3.9}$$

Each new point x' is then classified by evaluating:

$$y' = \text{sign}(\langle w^*, x'\rangle + b^*) \tag{3.10}$$

3.1.2 Soft-margin Linear SVMs

The type of SVMs detailed in the last section is called hard-margin separable SVM classifiers. In case the training data can not be separated without error, there is a variant called soft-margin SVMs, which allow misclassification with some penalty factor. If there is no hyperplane that can split the two classes, then soft-margin SVMs will choose a hyperplane that splits the examples as clearly as possible with some errors. This is achieved by introducing a slack variable ξ_i, which allows the possibility of examples violating constraints of hard margin SVMs. The quadratic optimisation problem becomes:

$$\min_w \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}\xi_i \tag{3.11}$$

subject to:

$$\begin{aligned} y_i(\langle w, x_i\rangle + b) &\geq 1 - \xi_i \\ 1 \leq i &\leq n \\ \xi_i &\geq 0 \end{aligned}$$

where C is the penalty factor which controls the trade-off between maximising the margin and minimising the classification error. The dual of the above equation can be written as:

$$L_D(w, b, \alpha, \xi) \equiv \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\langle w, x_i\rangle + b) - 1 + \xi_i) \tag{3.12}$$

Differentiating the above equation w.r.t. w, b and ξ and setting the derivative to zero, we get the following optimisation problem:

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{3.13}$$

3.1.3 Non-linear SVMs

Real world problems are not always linearly separable. SVMs very successfully utilise the 'kernel trick', which is a method for using a linear classifier to solve non-linear problems. This is done by mapping the original non-linearly separable data points into a higher dimensional space, where a linear classifier can be used. This makes the linear classification in high space equivalent to non-linear classification in the original space. The optimisation problem given in Equation 3.8 becomes:

$$L_D(w, b, \alpha) \equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i \tag{3.14}$$

where $K(x, x')$ is the kernel function doing non-linear mapping into the feature space such that:

$$K(x, x') = \langle\phi(x), \phi(x')\rangle \tag{3.15}$$

where $x, x' \in \mathcal{X}$ and ϕ is a mapping function such that $\phi : \mathcal{X} \to F$. The input space is defined by $\mathcal{X}$ and the feature space is

$$F = \{\phi(x) : x \in \mathcal{X}\} \tag{3.16}$$

However, the 'trick' is that by using kernels, we never explicitly compute the ϕ function. Kernel function (K) computes the inner product $\langle\phi(x), \phi(x')\rangle$ in feature space directly as a function of input space $\langle x, x'\rangle$. Commonly used kernel functions are linear kernels,

Gaussian radial basis function (RBF) and some sigmoid functions. This kernel property makes SVM a state-of-the-art classifier for machine learning problems.

In all cases reported in this thesis, we have used one-vs-one structure for extending binary SVMs to multiclass classification problem. In Chapter 6, we have tested several other structures for this task. We have used the LibSVM software library (http://www.csie.ntu.edu.two/~cjlin/libsvm/) to realise these classifiers. At each node, a linear SVM classifier has been used. We use a linear kernel instead of, say, a radial basis function (RBF) kernel since, according to Hsu *et al.* (2003), when the number of features is very large compared to the number of instances, there is no significant benefit to using an RBF kernel over a linear SVM.

There is a performance gain to be achieved by trying to optimise the value of C in the linear SVMs. However, in all of our initial experiments, best results were obtained for small values of C. Therefore we have decided to fix this value to $C = 0.1$ for all of the tests.

3.2 Validation Methods

When any dataset is made available for public use, training and test data partitions are not defined. Therefore, to obtain the generalisation performance, one has to apply some validation methods. The whole dataset must be divided into the training and testing parts. All the training of the classifiers is done only on the training part of dataset and the trained models are then tested on the testing part. In this way the trained models have no information about the test data. If this partitioning is unclear, then there is high chance of reusing the training data for testing, which will make the results overly optimistic and invalid. Usually there are two types of validation methods used in emotional speech recognition: k-fold cross validation and leave-one-speaker out cross validation.

3.2.1 $k-$Fold Cross Validation

To obtain the generalisation error on the dataset, we have applied stratified k-fold cross validation (CV) for splitting each database into k-folds (Witten and Frank, 2005). The labelled database is randomly divided into k disjoint subsets ('folds') where each fold contains approximately the same proportions of the emotion classes as the original dataset. Assuming the original data has n instances, then each fold will contain approximately n/k instances. During the learning and classification process, k multiclass classifiers are generated using learning algorithms; for each, one of the k folds is left out of the training set and used only for testing the trained models. The process is repeated k times, each time using a different fold for testing and the remaining $(k-1)$ folds for training. The

performance of the classifier is then taken as the average over the k runs. The most common value of k used in the literature is 10; therefore, this value has been adopted in our work.

3.2.2 Leave-One-Speaker Out Validation

A problem of randomly splitting the database into k-folds is that it does not guarantee speaker independence; data from the same speaker are likely to appear in both training and test data. This does not reflect the real-world scenario in which the classifiers generally do not have any information about the test speaker. Hence, the results of such a 'speaker-dependent' cross validation (SD–CV) will be over-optimistic.

To get more realistic results, one should perform 'speaker-independent' cross validation (SI–CV). For a total of s speakers in the whole dataset, the labelled data is divided into s disjoint folds where each fold contains the data from only one speaker, hence leave-one-speaker out cross validation. During learning and classification, s classifiers are generated; for each, one of the s speakers is left out of the training set and their data used only for testing.

It is understood that SI–CV is a much more difficult problem as compared to SD–CV as the trained models have absolutely no information about test speakers. Hence the classification accuracy for SI–CV should be lower in comparison to SD–CV. In such situations one may apply some sort of adaptation method on the extracted features or on the trained models. We will discuss this topic in further details in Chapter 7.

3.3 Evaluation Measures

In this section, we explain the evaluation measures used to present the results of classification studies in this thesis. This section has been included because there were two different measures used in the Interspeech 2009 Emotion Challenge (Schuller *et al.*, 2009b) and subsequent Interspeech 2011 Speaker State Challenge and Audio/Visual Emotion Challenge (AVEC) (Schuller *et al.*, 2011a).

In the Interspeech 2009 Emotion Challenge, weighted and unweighted average recall and precision were used as the performance measures. In subsequent challenges, only weighted and unweighted average recall were used as the performance measures. Most of the papers in this domain use weighted average accuracy as the performance measure to report their results. In this thesis we present our results as weighted and unweighted average accuracies. In the subsequent sections we give the details of all of these measures with their mathematical formulations and explain the equivalence of weighted/unweighted recall and weighted/unweighted accuracies.

Input		C_1	C_2	C_3	$\ldots$	C_M	$\sum$
	C_1	n_{11}	n_{12}	n_{13}	$\ldots$	n_{1M}	N_1
	C_2	n_{21}	n_{22}	n_{23}	$\ldots$	n_{2M}	N_2
	C_3	n_{31}	n_{32}	n_{33}	$\ldots$	n_{3M}	N_3
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	C_M	n_{M1}	n_{M2}	n_{M3}	$\ldots$	n_{MM}	N_M
	$\sum$						N

Figure 3.2: Sample confusion matrix for a M-class problem.

3.3.1 Weighted and Unweighted Average Accuracy

Traditionally, recognition rate or average accuracy (AA) is a widely used measure to report classification results. The AA measure indicates how many correct decisions a classifier has made. It is the sum of the correctly classified samples of each class (diagonal elements of confusion matrix shown in Figure 3.2) averaged by the total number of samples in the test set. Mathematically, it is given as:

$$\begin{aligned} \text{AA} &= \frac{n_{11} + n_{22} + \ldots + n_{MM}}{N_1 + N_2 + \ldots + N_M} \\ &= \frac{1}{N} \sum_{m=1}^{M} n_{mm} \end{aligned} \tag{3.17}$$

where M is the total number of classes, n_{mm} is the number of correctly classified samples of class m and N is the total number of test samples. This measure is also called weighted average (WA) accuracy as it is the ratio of correctly classified test samples per class weighted by the total number of test samples in each class and averaged by the total number of test samples in the test dataset.

The WA is a very good and widely used measure for reporting classification performance. Unfortunately, it is highly dependent on the class prior probabilities ($p_m = N_m/N$). A clueless classifier, which always predicts in the favour of only one class with maximum prior probability (p_m) independent of any other information given, has a total recognition accuracy of p_m. One can see that this will not be a problem when the dataset is balanced or reasonably balanced in terms of samples per class, but in the case of a highly imbalanced dataset, this measure can give unreasonably high average accuracy results. As an example, consider a two class (binary) problem in which 80% of the test data belongs to one class say 'neutral'. Our clueless classifier will predict all of the test samples as belonging to this class, achieving 80% WA. Thus high WA in the case of imbalanced dataset can be misleading.

To cater for imbalanced data, we also present our results as unweighted average (UA) accuracy. Mathematically, UA is given as:

$$\begin{aligned} \text{UA} &= \frac{n_{11}/N_1 + n_{22}/N_2 + \ldots + n_{MM}/N_M}{M} \\ &= \frac{1}{M}\sum_{m=1}^{M}\frac{n_{mm}}{N_m} \end{aligned} \tag{3.18}$$

If we consider the same imbalanced example with 80% test data belonging to *neutral* class and a clueless classifier, the UA in this case will be 50% which is a much better representation of the true performance of the classifier.

3.3.2 Recall and Precision for a Multiclass Problem

Recall and precision are the two measures which are used in conjunction for presenting the results on an imbalanced dataset. Recall is the fraction of test samples from class m correctly classified as m while precision is the fraction of total test samples classified as class m which actually belong to class m. They are defined as follows for a single class m:

$$\begin{aligned} \text{Recall}_m &= \frac{n_{mm}}{\sum_{i=1}^{M} n_{mi}} \\ &= \frac{n_{mm}}{N_m} \end{aligned} \tag{3.19}$$

$$\text{Precision}_m = \frac{n_{mm}}{\sum_{i=1}^{M} n_{im}} \tag{3.20}$$

In the Interspeech 2009 Emotion Challenge, weighted and unweighted average recall and precision were used as the performance measures. In a subsequent Interspeech 2011 Speaker State Challenge and the Audio/Visual Emotion Challenge (AVEC) 2011, only weighted and unweighted average recall were used as the performance measures. The weighted (WAR) and unweighted average recall (UAR) are defined as:

$$\begin{aligned} \text{WAR} &= \frac{(n_{11}/N_1 \times N_1) + (n_{22}/N_2 \times N_2) + \ldots (n_{MM}/N_M \times N_M)}{N_1 + N_2 + \ldots + N_M} \\ &= \frac{1}{N}\sum_{m=1}^{M} n_{mm} \end{aligned} \tag{3.21}$$

$$\begin{aligned} \text{UAR} &= \frac{n_{11}/N_1 + n_{22}/N_2 + \ldots + n_{MM}/N_M}{M} \\ &= \frac{1}{M}\sum_{m=1}^{M}\frac{n_{mm}}{N_m} \end{aligned} \tag{3.22}$$

As can be seen from the equations, WAR = WA and UAR = UA respectively. Hence in this thesis we present our results as UA and WA to compare our results with the published results.

Obviously, UA and WA accuracies for a perfectly balanced multiclass problem are going to be identical. However, for an unbalanced multiclass problem, UA gives a much more realistic performance. Therefore, we have given the results of classification performance as UA and we have only given WA where required for the comparison with the published research.

3.4 Databases

In Section 2.5 we discussed the different types of databases that are being used in current research. In this section we give details of the four databases we have used in our work. Out of the selected databases, three contain acted speech while one contains the spontaneous speech.

3.4.1 Danish Emotional Speech Database

The Danish emotional speech (DES) database is described by Engberg and Hansen (1996). It is only available for non-commercial research use. DES was recorded in Aarhus Theatre for Center for Person Komunikation (CPK), Aalborg University, Denmark in 1995. Four professional speakers, 2 males and 2 females, were asked to speak predefined sentences and words in Danish for 5 emotions: *neutral*, *angry*, *happy*, *sad* and *surprised*. Each speaker was asked to say 2 words, 9 short sentences and 2 passages ('paragraphs') in all 5 emotions. The average length of spoken words is 1 s; the sentences consist of on average 4.5 words lasting for 1.5 s. The paragraphs consist of 2 and 4 sentences each lasting for 10 s and 26 s, respectively. A total of 260 sentences is available in the database, with 52 sentences per emotion class making up 28 minutes of speech material. All recorded samples were included in the database. The quality of the acted emotions was verified by 20 human listeners, who were allowed to listen to them as many times as they wished before classifying them into one of the five emotion classes. This revealed that the *neutral* emotion is very strongly confused with *sad*; *angry* with *neutral* and *surprised*; *happy* with *neutral* and *surprised*; and *surprised* with *happy* and *neutral*. Reported human accuracy on this database is 67.3%.

Other researchers have treated the two passages differently. Sometimes they are left out of the training and testing sets, whereas in other cases they are divided into sentences (by detecting inter-sentence pauses) leading to a database consisting of over 400 sentences. In our work, to keep things simple and make future comparisons easier, we have omitted the passages.

3.4.2 Berlin Database

The Berlin database, also known as Emo-DB, contains utterances spoken in German. It is available at http://pascal.kgw.tu-berlin.de/emodb/index-1024.html (last visited 20 February 2012). The database was recorded in 1997 and 1999 in an anechoic chamber at the Technical University, Berlin. Ten professional native German actors, 5 males and 5 females, were asked to speak 10 sentences in 7 different emotions: *neutral*, *anger*, *happiness*, *sadness*, *fear*, *boredom* and *disgust*. Note that four of these classes are common with DES. These sentences were then evaluated by 20–30 listeners to verify the emotional state and only those were retained that had a recognition rate of 80% or above and were judged as natural by more than 60% of the listeners, yielding "about 500 utterances" in total making up 22 minutes of speech material. Each sentence consists of on average 10 words with average duration of approximately 5 s. Reported human accuracy on this database is 86.1%.

3.4.3 Serbian Database

The Serbian database of acted emotional speech (Jovicic *et al.*, 2004) was recorded in 2003 in an anechoic studio at the Faculty of Dramatic Arts, Belgrade University, Serbia, using 6 actors: 3 males and 3 females. It has been less well used than DES and Berlin. It consists of 32 isolated words, 30 short semantically-neutral sentences, 30 long semantically-neutral sentences and one passage consisting of 79 words, i.e., $32 + 30 + 30 + 1 = 93$ utterances. The following 5 emotions are represented: *neutral*, *anger*, *happiness*, *sadness* and *fear*. Hence, there are $93 \times 6 = 558$ sentences per emotion. Each of the 93 utterances is contained in a separate `.wav` file; so there are $93 \times 6 \times 5 = 2790$ files in total. Each speaker was recorded in separate sessions so that they do not influence each other's speaking style. Each recorded sentence was evaluated by 39 listeners; reported human accuracy on this database is 94.7%. In general, these human listening tests show that *anger* and *happy* emotions are often confused with each other, whereas *neutral* is most usually confused with *sad*.

3.4.4 Non-Acted Speech: AIBO Database

The Aibo database (Batliner *et al.*, 2004; Steidl, 2009) contains recordings of children interacting with Sony's pet robot Aibo. It consists of induced, German speech that is emotionally coloured. The children were led to believe that Aibo was responding to their commands, whereas it was actually controlled by a human operator in a Wizard-of-Oz manner. Sometimes Aibo behaved 'disobediently', thereby provoking emotional reactions. The data were collected at two different schools, identified as 'Mont' and 'Ohm', with 25 and 26 children speakers from each, respectively. Five expert human

Table 3.1: Emotion classes and number of sentences per class for (a) acted DES, Berlin and Serbian databases; (b) the spontaneous Aibo corpus. The horizontal line in (a) separates the emotions that are common to all three acted databases from those which are not. See text for explanation of 'Mont' and 'Ohm'.

(a) Acted speech data

DES	**Sentences**	**Berlin**	**Sentences**	**Serbian**	**Sentences**
Neutral	52	Neutral	79	Neutral	558
Angry	52	Anger	127	Anger	558
Happy	52	Happiness	71	Happiness	558
Sad	52	Sadness	62	Sadness	558
Surprised	52	Fear	69	Fear	558
		Boredom	81		
		Disgust	46		
Speakers	2M,2F		5M, 5F		3M, 3F

(b) Spontaneous speech data

	Sentences	
Aibo	**Mont**	**Ohm**
Anger	611	881
Emphatic	1508	2093
Neutral	5377	5590
Positive	215	674
Rest	546	721
Speakers	25	26

labellers listened to the speech data and annotated each word independently. They were asked to put the 48,401 words in the database into the following ten categories: 'angry', 'touchy'/'irritated', 'joyful', 'surprised', 'bored', 'helpless', 'motherese', 'reprimanding', 'emphatic', and a category 'other' for all remaining cases which were rare and not covered by the other classes. For subsequent machine-learning purposes, the 10 classes were further mapped to 4 classes: *anger*, *emphatic*, *neutral* and *positive* with a fifth category for *rest*. Word labels were then mapped to so-called 'turn' (i.e., utterance-level) labels using a heuristic method described by Steidl (2009), to make up a total of 18,216 sentences. Speaker independent recognition can be achieved by using recordings from one school for training and the other for testing. The Aibo corpus formed the focus of the recent Interspeech 2009 Emotion Challenge (Schuller *et al.*, 2009b). Table 3.1 shows details (number of sentences and emotions covered) of the four databases that we have used in this work. These are the acted DES, Berlin and Serbian databases, and the spontaneous Aibo corpus.

3.5 Summary

In this chapter we have given a brief introduction to SVM classification algorithms. This mathematical description will be very helpful in the later chapters where we derive the mathematical formulation for weighted-SVMs. We have also described the evaluation methods which are used to get the generalisation performance of the classifier, namely the SD–CVand SI–CV validation methods. The performance measures used in this thesis are given in detail. Finally, acted and spontaneous emotional speech databases used in this thesis have been outlined.

Chapter 4

Features for Emotion Recognition from Speech

A very important issue in the design of an SER system is the extraction of suitable features that can correctly and efficiently characterise different emotions. So far a large number of different features have been proposed for recognising emotion-related user states. These features can be characterised into linguistic, acoustic and semantic features. In this thesis, we have used acoustic features only for emotion recognition.

This chapter details the current issues in feature extraction faced by researchers. Different types and categories of features are given along with the details of the extraction methods used. We show and explain how these features are affected by different emotions. The details of the state of the art acoustic feature set used in our work are given at the end of this chapter.

4.1 Current Issues in Feature Extraction

There are three main issues that need to be considered and addressed in feature extraction. The first is choosing an appropriate unit for analysis and representation for feature extraction. Some researchers divide the speech sample into intervals, called frames, and extract the local features from each frame while others follow Murray and Arnott (1993) and use only the global statistics over the whole utterance or use several frames as a single unit or segment for feature extraction. Another common yet unresolved issue is determining the best acoustic feature for emotion recognition from speech out of prosodic, spectral and voice quality features. The last issue related to feature extraction is whether using only acoustic features are enough for modelling emotions or whether other types of features, like linguistic, facial expression or discourse information need to

be included in the feature set. We shall now discuss all of the above mentioned issues in detail.

4.1.1 Feature Representation

A speech signal is traditionally analysed by dividing it into short frames of 20 ms in length with an overlapping window of 10 ms. By doing so, the signal within a frame can be considered as approximately stationary and signal processing techniques can be applied to it. Features like pitch, energy, formants, MFCC and voice quality, are extracted from each frame. These can be called local features. Some researchers call them low-level descriptors (LLD) (Schuller *et al.*, 2007b) and use them directly for modelling and testing the classifiers on the frame level. After extracting the LLDs, global statistical functionals like max, min, standard deviation, can be calculated over several frames or even the whole utterance. These functionals give the global trends over the whole utterance. The majority of researchers use these global features and advocate their superiority over local features in terms of classification accuracies.

In the literature there are different terms used for the extracted features based upon the unit of speech sample used for extraction features. Long-term versus short-term features, local versus global features and intra versus supra segmental features are some of the terms consistently used in the literature. In this section, we will explain these terms to clarify their differences.

4.1.1.1 Long-term Features

Dellaert *et al.* (1996) and Murray and Arnott (1993) are the pioneers in the area of emotion synthesis and recognition from speech using machine learning techniques. They started their work by utilising recorded speech from actors. As the text and emotions to be portrayed were predefined, they assumed that the emotion will remain constant over the whole utterance and treated it as a single unit. Global features were calculated from the LLDs by utilising the whole utterance. The trend of using the whole utterance as a one single unit and utilising feature extracted over the whole utterance for classification has been accepted as the basis of most subsequent studies. Therefore, a complete sentence or a turn is considered at a time and all the features are extracted from it. These global features are also called *long-term* (Li and Zhao, 1998). Sometimes, these features are also referred to as supra-segmental features by Yang and Lugger (2010).

Most research work in emotion recognition from speech has relied upon these long-term features. In this case, only a single value is calculated for a whole utterance. This representation captures only the global trend while the temporal information present in the speech signal is completely lost. Researchers like Nwe *et al.* (2003) have claimed

that long-term features are only useful in discriminating between the high versus low arousal emotions. They claim that these features are not very successful in separating those emotions which have the same arousal level like angry and happy emotions for which one has to concentrate on short-term features.

4.1.1.2 Short-term Features

Researchers who work on spontaneous speech question the approach of Murray and Arnott (1993) and argue against considering a complete utterance as a single unit for extracting features. As the emotions in spontaneous speech can be short-lived, the speech features should perhaps be calculated over smaller segments than utterances. Researchers have tried to use the features extracted from each frame as an input feature to the classifier. Schuller *et al.* (2009a,b) have used both frame level modelling with HMM classifiers and utterance level modelling using global features with SVMs on several of the databases mentioned in Table 2.2. In all of their tests, they found that global features outperform frame level feature modelling. The difference between the performance of the two approaches is very large for acted databases but not as large for spontaneous databases.

Another approach is to extract features from segments of speech consisting of phones. This approach is based upon the study by Lee *et al.* (2004) in which they observed variations in the spectral shape of the same phone in different emotions. This observation is only true for vowel sounds and secondly, it will be severely affected by a poor phone segmentation algorithm, especially when the phonetic transcriptions are not available.

A third approach used by Shami and Verhelst (2007), Casale *et al.* (2008) and Shaukat and Chen (2008) is to break down each speech sample into voiced and unvoiced segments. Acoustic features are extracted from these short voiced segments while the unvoiced segments are only used to measure few voice quality features, hence referred to as *segmental* features. Shami and Verhelst (2007) and Shaukat and Chen (2008) have used the length of the segment to weight the posterior class probabilities and the final decision is made as a weighted sum for all of the segments in the speech sample. In their experiments, all of the authors report that utterance level features outperform segmental features. Similarly, Steidl (2009) found that extracting features from a self-defined chunk level performed best on the FAU Aibo emotional speech database. These chunks were larger than a word but smaller than a turn or a whole sentence.

4.1.2 Categories of Acoustic Features

An important issue in any SER is to extract speech features that can efficiently characterise emotion from speech while not depending heavily on the speaker. Although many

features have been explored by researchers, the best ones for SER systems have still not been identified. In this section we will introduce the acoustic and linguistic features utilised by the researchers. The acoustic features can be categorised into prosodic, spectral and voice quality features.

Prosodic Features

Since the often cited work of Murray and Arnott (1993), prosodic features are the most commonly used set of features in emotion recognition. Prosodic features are supra-segmental, i.e., they are not confined to phoneme-like small segments, rather they characterise speech segments like words, phrases or whole utterance. The prosodic features can be categorised into pitch, energy and speaking rate. These features characterise the properties of 'source' in the source-filter model of human speech production system.

The contours of pitch and energy are affected by the physiological changes in the current state of the speaker. After estimating the contour, several functionals like mean, median and standard deviation, are calculated to get an overall effect of specific emotions on the contour. Often functionals are extracted from the contour directly as well as the first derivative ($\triangle$ features) and second derivative ($\triangle\triangle$ features) of the pitch and energy contours. Speaking rate based features model the effect of the speaker's current state on the duration of the spoken words. They can be measured on various units like a single word or the whole utterance.

Spectral Features

Spectral features describe the characteristics of the speech signal in the frequency domain besides fundamental frequency f_0. One of the most commonly used spectral features of speech are the formant frequencies. Formants are the frequencies which result from resonance in the vocal tract. When we speak, the vocal tract is constantly being modified to articulate the speech. The length of the vocal tract is affected by the emotional state of the speaker which in turn affects the formants. The formants are characterised by their centre frequencies, amplitude and bandwidth. The voiced part of the speech can have four or more formants. In practice only first two formants along with their bandwidths are used for emotion recognition. The application of formants in emotion recognition is demonstrated in several studies.

Standard mel-frequency cepstral coefficients (MFCC) used in automatic speech recognition systems (SER) have also been found useful in emotion recognition. Although they were originally designed to recognise 'what' is being spoken, they have proven to be useful in recognising 'how' something is being spoken (Kim *et al.*, 2007; Shami and Verhelst, 2007; Shaukat and Chen, 2008; Rong *et al.*, 2009). These features can be either

used for classification at the frame level (short-term features) like Kim *et al.* (2007) or averaged over a larger segment like word or utterance (long-term features) like Shaukat and Chen (2008).

Some authors like Eyben *et al.* (2009) argue that instead of using MFCC coefficients in which most of the data is not used, the output of the mel-filter bank can be directly used for extracting features. These signals should have much more information than the MFCC. These features can be used along with MFCC and other spectral features.

Voice Quality Features

Voice quality characterises the speech into breathy, whispery, creaky, harsh voice. According to the source-filter model, voice quality features are mainly related to the variations in the voice source or glottal excitation. Usually, inverse filtering is applied to the speech signal to separate the source signal. However this method is very sensitive to noise. An alternate to inverse filtering is to measure the source related properties directly from the spectrum of the speech signal. The two most popular voice quality features are jitter and shimmer. Jitter and shimmer are cycle-to-cycle variations of fundamental frequency and waveform amplitude respectively.

Voice quality features used by Fernandez and Picard (2005) and Planet *et al.* (2009) can be considered as classical features for detecting emotions from speech. Recently, Yang and Lugger (2010) proposed their novel voice quality features and found them to provide competitive results on the Berlin database.

Linguistic Features

Every language has some particular words to express a specific emotional state, e.g., if a person is in a happy state, in English he or she might use words like, *great*, *fantastic* and *brilliant*. Such linguistic features are a rich source of information for determining the current state of the speaker. However, usually when databases such as DES, Berlin and Serbian are designed, one of the techniques used is to employ non-emotional text in order to make the analysis of acoustic features easier. Hence, linguistic features are often not utilised in SER systems.

A common approach is to estimate the class probability based upon the sequence of words like in language models. Instead of using n-gram language models, usually a unigram model is used for emotion recognition. Lee and Narayanan (2005) used a unigram model to detect the 'salient' words from a sentence. A new measure *emotional salience* was defined which is the measure of how much information a word provides towards signalling certain emotions. Acoustic and language information was used to recognise negative emotions (anger and frustration) from a commercially deployed call

centre application. Reported classification accuracy using a k-NN classifier with acoustic features only was 78.6% and with addition of language features was improved to 86.4%. Yildirim *et al.* (2011) have extended the idea of Lee and Narayanan (2005) from single words to pair of words and phrases. They have used the 384 acoustic features as were proposed in Schuller *et al.* (2009b) and apply them on a spontaneous database consisting of child–machine spoken dialogue interaction in a game setting.

A second approach is spotting emotional keywords in the utterances that give the information about the current emotional state of the speaker. Emotional keyword spotting based on Bayesian belief networks was proposed by Schuller *et al.* (2004). An emotional speech database consisting of English and German sentences was collected. It contained 7 emotional classes and SVMs were chosen as the acoustic classifier. By combining the linguistic information with the acoustic information, the classification accuracy was increased from 74.2% using only acoustic features to 83.1% using both.

4.1.3 Combining Acoustic Features with other Information Sources

There are several sources of information that have been augmented with acoustic features for recognising emotions. As mentioned in the previous section, linguistic information is one of the sources that can be used. Discourse information was used by Lee and Narayanan (2005) for detecting negative emotions from real users engaged in spoken dialogue with a machine agent over the telephone using a commercially deployed call centre application. Often the automated agents do not perform to the satisfaction of the customers and in response they express negative emotions like *frustration* or *anger*. They found a strong correlation between the speech-act of rejection and negative emotions. They combined acoustic, language and discourse information and reported 85.1% WA accuracy using a k-NN classifier and floating forward selection as the feature selection method.

Another source of information is the video information which has quite recently come to the attention of researchers. Recently held Audio/Visual Emotion Challenge (AVEC) 2011 is one of the examples of research in this area. Haq and Jackson (2009) have performed some audio-visual emotion recognition experiments in which audio and visual features were extracted independently. Sequential floating forward selection was used to select the final feature set from the combination of audio and visual features. They report that out of many fusion methods tried, fusion of audio and video features after feature reduction performed best.

4.2 Feature Extraction Methods

In this thesis we have concentrated only on the acoustic features extracted from the speech samples. Table 4.1 highlights some of the acoustic feature sets used by other researchers in this domain. Most researchers have used prosodic and spectral features. This list is not comprehensive but gives the general idea about different feature sets being used. Methods for estimating the acoustic features are described hereafter. All acoustic features are estimated on a frame basis with each frame of length 20 ms with overlapping window of 10 ms on both sides. We have mainly concentrated on these proven acoustic features which work well for emotion recognition.

4.2.1 Fundamental Frequency/ Pitch

From the source filter model discussed in Chapter 2, sound is produced when pressurised air from the lungs passes over the tense vocal folds. The rate at which these vocal folds vibrate determines the fundamental frequency (f_0) or pitch of the produced speech. It has been observed by many researchers that human emotions have a strong effect on the contours of the pitch and energy carried in the sound. Changes in the fundamental frequency contour has been used extensively for emotion recognition as can be seen in Table 4.1.

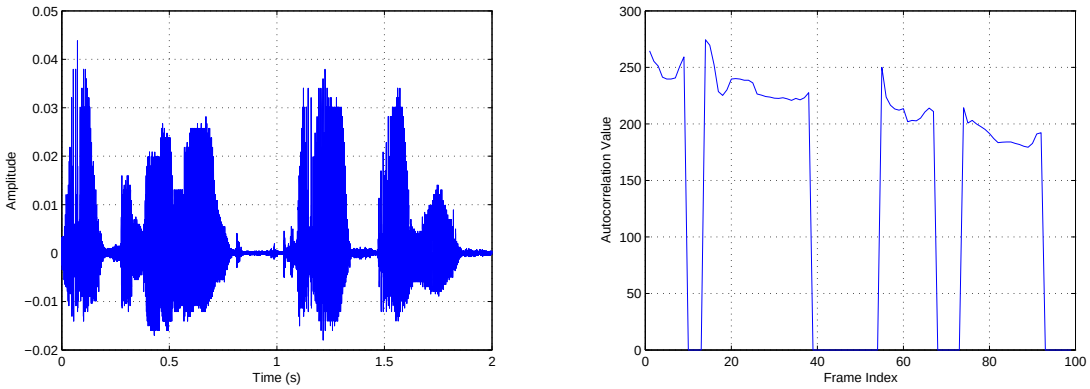
Pitch detection is a difficult topic in speech signal processing. However, over the course of time, there have been several very stable pitch detection algorithms proposed. We can divide f_0 estimation algorithms into two major groups, which are time-domain and frequency-domain methods. Each method has its advantage and disadvantage depending upon the specific requirement of the application. Boersma (1993) proposed an autocorrelation based method which is considered as an accurate way for estimating the pitch contour from speech. Figure 4.1(b) shows the pitch contour of the speech sample shown in Figure 4.1(a) using this method. Some other methods that were explored for f_0 estimation from time domain and frequency domain are explained in Appendix A.

Figure 4.2 shows the distribution of mean pitch (f_0) values in five different databases calculated over the whole utterance. An obvious observation can be made from the plots that for acted databases, the difference between average f_0 of different emotions is very obvious as compared to the spontaneous databases (Aibo-Mont and Aibo-Ohm). On average, *happy* and *angry* emotions have higher average f_0 values in comparison to *neutral* and *sad* emotions. This difference is very clear for Berlin and Serbian databases, both of which contain acted database. However, it is not so clear in DES and Aibo-Ohm databases.

One would expect that for the acted emotional speech database DES, this difference should have been large as the emotions are portrayed by actors and should have been

Table 4.1: Summary of the acoustic features used by different researchers.

Reference	Pitch	Energy	Rhythm	Spectral	Formants	Voice Quality
Nwe *et al.* (2003)	-	✓	-	✓	-	-
Oudeyer (2003)	✓	✓	-	✓	-	-
Lee and Narayanan (2005)	✓	✓	✓	-	✓	-
Fernandez and Picard (2005)	✓	✓	-	-	-	✓
Ververidis and Kotropoulos (2006)	✓	✓	-	✓	✓	-
Batliner *et al.* (2006)	✓	✓	✓	✓	✓	✓
Kim *et al.* (2007)	✓	✓	-	✓	-	-
Shami and Verhelst (2007)	✓	✓	✓	✓	-	-
Shaukat and Chen (2008)	✓	✓	-	✓	-	✓
Rong *et al.* (2009)	✓	✓	✓	✓	-	-
Schuller *et al.* (2009a)	✓	✓	✓	✓	✓	✓
Xiao *et al.* (2010)	✓	✓	-	-	✓	-
Yang and Lugger (2010)	✓	✓	✓	-	✓	✓
Eyben *et al.* (2010)	✓	✓	✓	✓	✓	✓
Schuller *et al.* (2011a)	✓	✓	✓	✓	-	✓
Lee *et al.* (2011)	✓	✓	✓	✓	-	✓
Yildirim *et al.* (2011)	✓	✓	✓	✓	-	✓



(a) Speech signal of a female speaker.

(b) Autocorrelation contour using threshold of 0.38.

Figure 4.1: (a) Speech of a female speaker with some voiced and unvoiced sounds. (b) Pitch contour using autocorrelation method of Boersma (1993).

exaggerated to an extent. However, there is not a large difference between mean f_0 for all of the emotions. One can assume that the emotions are not very well portrayed which is why the average human accuracy reported on this database is 67.3% (Engberg and Hansen, 1996). For the spontaneous database, Aibo-Ohm one can expect this difference to be not very large as the emotions are spontaneous and not full-blown.

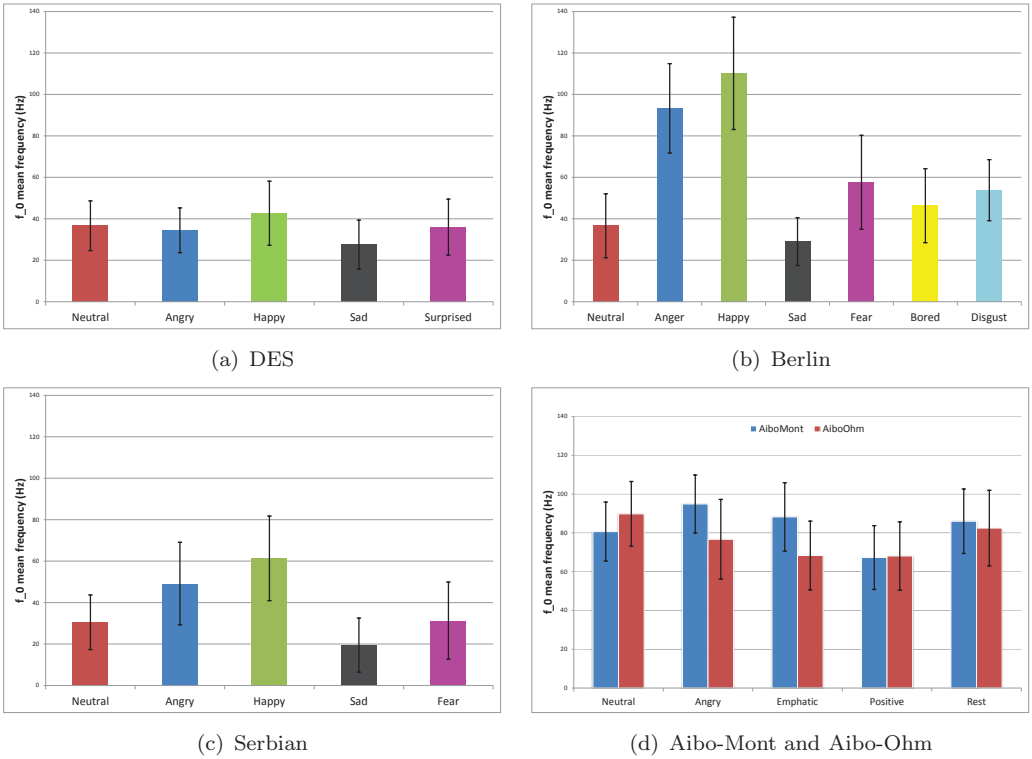
4.2.2 Energy Features

Energy in the speech signal comes from the pressure built up by the lungs and passed over the vocal folds and through the vocal tract. Energy in the speech signal has been found by many researchers to be very useful in determining the emotions. In this regard, features utilised by Oudeyer (2003) stand out; he used 120 energy related features. Nwe *et al.* (2003) found log energy features to very helpful in separating six emotions. Fernandez and Picard (2005) have also proposed 20 intensity features based upon the Bark scale. Eyben *et al.* (2009) have proposed to use energy in different frequency bands. These features have been found to work quite well under different conditions for detecting emotions. For a signal of infinite extent, energy in that signal can be calculated by:

$$E_a = \int_{-\infty}^{\infty} |x(t)|^2 \, dt$$

For short-time discrete signals, the above equation can be written as:

$$E_d = \sum_{n=0}^{N-1} |x(n)|^2$$



(a) DES (b) Berlin

(c) Serbian (d) Aibo-Mont and Aibo-Ohm

Figure 4.2: Bar plots of average f_0 value per emotion along with the corresponding standard deviations for (a) five-class DES database; (b) seven-class Berlin database; (c) five-class Serbian database; (d) five-class Aibo-Mont and Aibo-Ohm databases.

Similarly, the short-time power of a signal can also be used instead of the short-time energy of a signal. The short-time power of a signal is given by:

$$P_d = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2$$

which we can see is quite similar to the expression of energy other than the $1/N$ factor which is the averaging factor over a signal of length N. These two features are used interchangeably. We have used power contour and its properties in this thesis. Figure 4.3 shows the power contour of the speech signal shown in Figure 4.1(a).

4.2.3 Duration Features

Human sounds can be broadly divided into two categories: voiced and unvoiced. The ratio of average length of voiced speech segment to the unvoiced segment gives the information about the speaking rate of the current speaker. This speaking rate can give some information about the current state of the speaker.

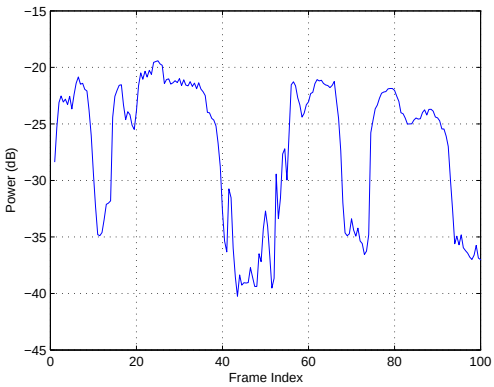


Figure 4.3: Power contour of the signal shown in Figure 4.1(a)

Vowels are usually categorised as voiced sounds and they often have high average energy levels and very distinct resonant or formant frequencies. Voiced sounds are generated by air from the lungs being forced through the adducted vocal cords. As a result the vocal cords vibrate in somewhat periodic pattern which determines the pitch of the voiced sound produced.

Consonants can be voiced as well as unvoiced sounds and generally have less energy and higher frequency components than voiced sounds. The production of unvoiced sounds involves air being forced through the vocal folds in a random flow. During this process the vocal folds do not vibrate, instead, they stay open and allow the air to pass through to the vocal tract. Pitch is a relatively unimportant attribute of unvoiced speech since there is no vibration of the vocal cords and no glottal pulses.

Several methods have been proposed for characterisation of speech into voiced, unvoiced and silence. This is one of the basic steps to do before performing feature extraction. The method proposed in Markel (1972) is based upon spectral flatness measure, energy in speech and zero crossing rate. Despite the several methods proposed for this task, they all rely basically on the same methods.

The spectral flatness makes use of the property that the spectrum of pure noise is expected to be flat. In other words, the spectrum of unvoiced speech is flat and the spectrum of voiced speech is less flat. The spectral flatness measure (SFM) for the jth frame is given by:

$$\begin{aligned} \text{SFM} &= \frac{G_m}{A_m} \\ &= \frac{\left(\prod_{n=0}^{N-1} X_j(n)\right)^{\frac{1}{N}}}{\frac{1}{N}\sum_{n=0}^{N-1} X_j(n)} \end{aligned} \tag{4.1}$$

where G_m is the geometric mean of the magnitude spectrum and A_m is the arithmetic mean of the magnitude spectrum. SFM ranges from almost 0.9 for a white noise to 0.1 for a voiced signal. The threshold is usually chosen to be $0.35 \sim 0.48$.

Zero-crossing rate (ZCR) is the number of times the signal crosses zero level threshold. ZCR for unvoiced sound is much higher than that of voiced sound. Rabiner and Sambur (1975) give in detail a method for voiced and unvoiced detection.

Similarly, Atal and Rabiner (1976) give their method for classification of voiced and unvoiced sounds in which they have based their classifier on five parameters. Along with energy and ZCR, they have used autocorrelation, LPC filter coefficients and energy in the residue signal. If the largest peak in the normalised ACF is less than a specific threshold, the frame is considered as unvoiced frame. Normally, a threshold of $0.38 \sim 0.42$ is used for this classification.

4.2.4 Spectral Features

In addition to prosodic features, spectral features are usually used as a short term representation of the speech signal. In order to exploit the human auditory system, the spectrum of the speech is passed through a bank of band-pass filters whose centre frequencies are based upon human perception scales. Spectral features are then extracted from these filtered signals.

Based upon the human auditory system, physiological studies model the human perception of speech as a bank of filters whose centre frequencies are approximately exponential. To model these frequencies, researchers have come up with two similar scales called Bark and mel scale. The Bark scale proposed by Eberhard Zwicker in 1961 is named after Heinrich Barkhausen who proposed the first subjective measurements of loudness. It is defined by

$$b(f) = 13 \arctan(0.00076 f) + 3.5 \arctan\left(\frac{f}{7500}\right)^2 \tag{4.2}$$

The second perceptually motivated frequency scale is the mel scale, which is approximately linear below 1000 Hz and logarithmic above. The mel scale is defined as:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{4.3}$$

Both Bark and the mel scale behave similarly and both have been used extensively for speech representation and recognition. There is a difference between the definition of the two scales, however they behave very similarly as shown in Figure 4.4.

Mel-frequency cepstral coefficients are standard features used in most SER systems. The reason that they are widely accepted is their relatively good performance and robustness

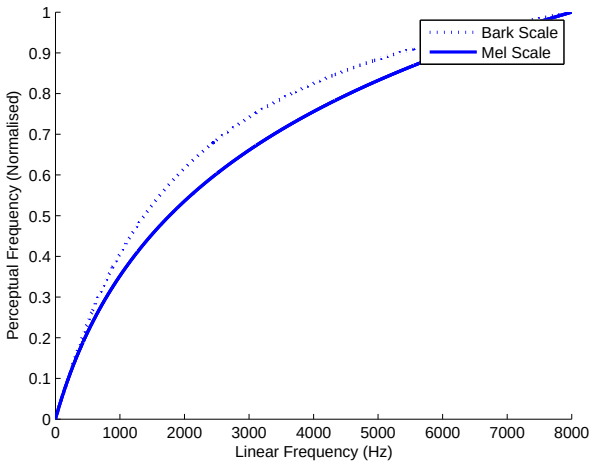


Figure 4.4: Bark and mel frequency scale.

to noise and environmental changes by adequate preprocessing. Usually 12 MFCC features are used along with the a log-energy feature making a total of 13 feature set.

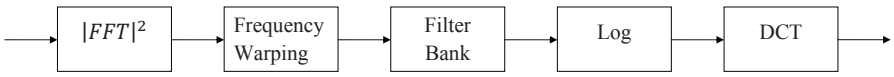
Figure 4.5(a) shows the block diagram of the steps involved in calculating the MFCC. Figure 4.5(b) shows an example filter bank where each one is a band pass filter. The benefit of using MFCCs is that a complete speech sample is compressed into only 13 coefficients. Kim *et al.* (2007) used GMMs to model MFCC features and combined the results with k-NN classifier applied on prosodic features. The results were reported in EER on a database collected for emotional speech research at USC. Similarly, Shaukat and Chen (2008) used the average value of MFCC features over the small voiced segments from the Serbian emotional speech database and reported 89% weighted average.

4.2.5 Formant Features

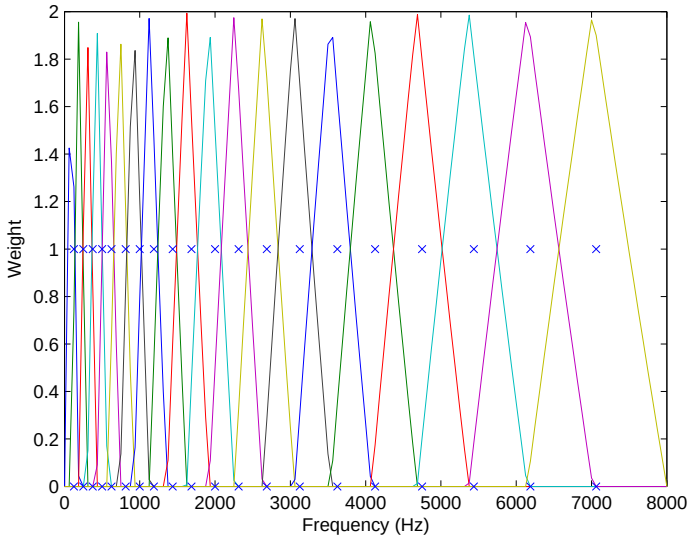
When we speak, the vocal tract is constantly modified to articulate the speech. The length of the vocal tract is also affected by the emotional state of the speaker. The frequencies at which the vocal tract resonate are the formant frequencies. They depend upon the shape and physical dimension of the vocal tract. Each formant is characterised by its centre frequency as well as its bandwidth. A simple method to model the vocal tract relies on linear predictive coding (LPC). It is modelled as a p-order all pole filter and coefficients are calculated by solving a system of linear equations. These are used to build the all-pole filter $\hat{H}(z)$ that estimates the response of vocal tract.

$$\hat{H}(z) = \frac{1}{1 - \sum_{i=1}^{p} \hat{a}(i) z^{-1}} \tag{4.4}$$

where $\hat{a}(i)$ are the linear predictive coefficients.



(a) Block diagram for calculating mel-frequency cepstrum coefficients (MFCC).



(b) MFCC filter-bank, crosses indicate the centre frequency of each filter.

Figure 4.5: (a) Block diagram for calculating MFCC. (b) MFCC filter-bank.

There are two methods for extracting formant frequencies and their bandwidth using LPA. For the first method, once the filter coefficients of $\hat{H}(z)$ have been calculated, the filter response can be used for determining the peaks where the filter response is maximum and these peaks give the formant frequency and their spread gives the bandwidth of that specific formant. These local maxima can be found by using any of the peak picking methods. This process involves calculation of the Fourier transform and then application of the peak picking method.

The second method involves computing formants directly by mathematical computations on filter coefficients $\hat{H}(z)$. It is known that the filter coefficients can be used to generate a polynomial whose degree will depend on the number of filter coefficients. The roots of this polynomial give us the location of poles in the z-domain. Therefore, if p is the number of filter coefficients, then we have $p/2$ complex conjugates. If $z_i = z_{ir} + jz_{ii}$ is the ith root, then from Snell and Milinazzo (1993) and Rabiner and Schafer (1978, Chap. 8) we find the following relations for formant frequency F and 3-dB bandwidth B:

$$F_i = \frac{1}{T} \tan^{-1}\left(\frac{z_{ir}}{z_{ii}}\right) \tag{4.5}$$

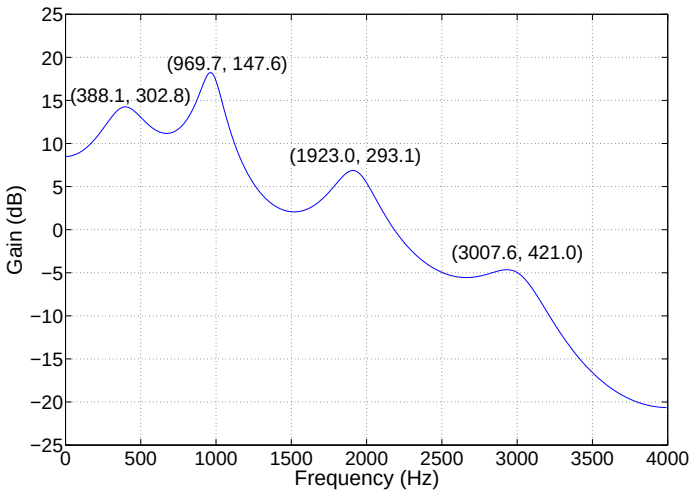


Figure 4.6: Fourier transform of a 20 ms frame of signal shown in Figure 4.1(a), peaks in the graph show the formant frequency locations. Numbers on the peak are the formant frequencies and their corresponding bandwidths.

and

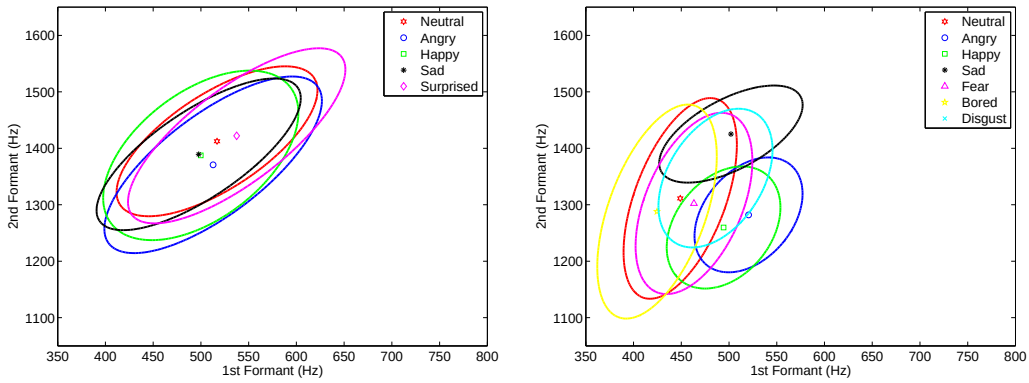
$$B_i = \frac{1}{2T} \log \left(z_{ir}^2 + z_{ii}^2 \right) \tag{4.6}$$

Figure 4.6 shows the response of the LPC filter for a 20 ms frame of the signal shown in Figure 4.1(a). The peaks are the locations of the first four formants and the numbers on the peaks are the corresponding formant centre frequency and their bandwidths calculated by using the above mentioned equations.

Accurately calculating the third and fourth formants in the presence of noise is not always possible. Therefore, only the first two formants and their bandwidths are used. Figure 4.7 shows the mean and contour plots for the distribution of the first and second formants for all the databases. The contours are plotted by using ellipses with major and minor axes equal to the variance of the data.

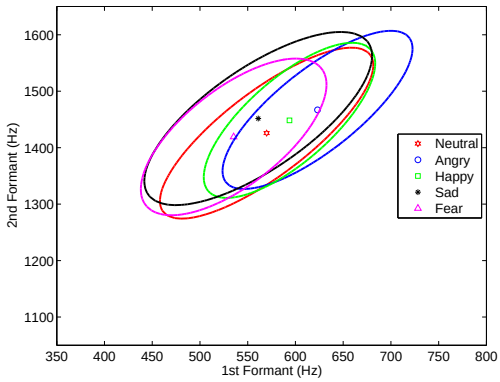
4.2.6 Voice Quality Features

Voice quality features characterise the properties of the source signal in the source-filter model. For this thesis, only those voice quality features are investigated which can be directly calculated from the speech signal, namely, jitter and shimmer and harmonics to noise ratio.

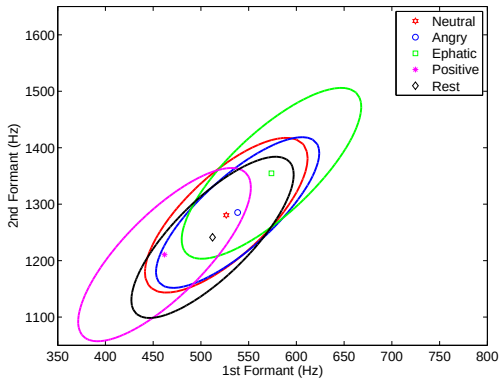


(a) DES

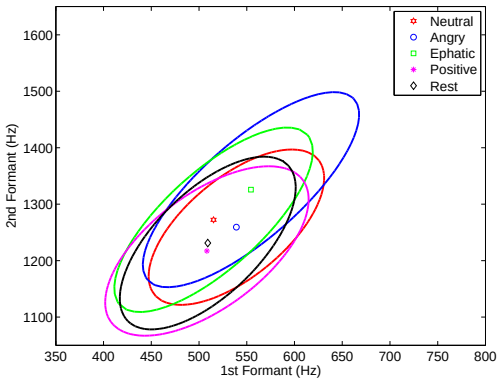
(b) Berlin



(c) Serbian



(d) Aibo-Mont



(e) Aibo-Ohm

Figure 4.7: Mean and contour plots for the distribution of 1*st* and 2*nd* formant for different emotions for (a) five-class DES database; (b) seven-class Berlin database; (c) five-class Serbian database; (d) five-class Aibo-Mont and (e) Aibo-Ohm databases.

4.2.6.1 Jitter and Shimmer

As mentioned earlier (Section 4.1.2) jitter is the cycle to cycle variations of the fundamental frequency (f_0) and is calculated as:

$$\text{jitter}(i) = \frac{|f_0(i+1) - f_0(i)|}{f_0(i)} \tag{4.7}$$

Similarly, shimmer is cycle to cycle variation in the energy. It is calculated as:

$$\text{shimmer}(i) = \frac{|E(i+1) - E(i)|}{E(i)} \tag{4.8}$$

4.2.6.2 Harmonics to Noise Ratio

The harmonic to noise ratio (Boersma, 1993) is the measure of the periodicity of a signal which can be found for a periodic signal by the relative distance between consecutive peaks in its autocorrelation function. Mathematically, the autocorrelation function (ACF) of a discrete signal $x(t)$ of infinite extent is defined by:

$$r_x(\tau) = \sum_{j=-\infty}^{\infty} x_j x_{j+\tau} \quad \tau = -\infty \ldots -1, 0, 1, \ldots, \infty \tag{4.9}$$

where $r(\tau)$ is the autocorrelation function of lag τ. For a signal consisting of W samples, the autocorrelation equation can be written as:

$$r_x(\tau) = \sum_{j=t}^{t+W-1} x_j x_{j+\tau} \quad \tau = \tau_{min}, \ldots, -1, 0, 1, \ldots, \tau_{max} \tag{4.10}$$

The function has a global maximum at the lag $\tau = 0$. If there is another maximum outside $\tau = 0$, the signal is called periodic and this information is used to calculate the pitch of a signal ($f_0 = 1/T_0$) where T_0 is the time period of the periodic signal. This method for detecting pitch is explained in further detail in Appendix A. The normalised autocorrelation function is given by:

$$r'_x(\tau) = \frac{r_x(\tau)}{r_x(0)} \tag{4.11}$$

The time period (T_0) is calculated by finding the distance between two consecutive peaks in r'_x. The next peak after $\tau = 0$ exists at the lag $\tau_{max} = T_0$. The value of ACF at $r'_x(\tau_{max})$ represents the relative power of the periodic signal.

If noise is added to the signal, then a framed speech signal consists of two parts: harmonic part $H(t)$ and noise part $N(t)$. Then the corresponding autocorrelation function at zero lag $\tau = 0$ also consists of two parts $r_x(0) = r_H(0) + r_N(0)$. The relative power of the

harmonic and noise signals at τ_{max} is given as:

$$r'_x(\tau_{max}) = \frac{r_H(0)}{r_x(0)}; \; 1 - r'_x(\tau_{max}) = \frac{r_N(0)}{r_x(0)} \tag{4.12}$$

and the harmonic to noise ratio (HNR) can be defined as:

$$\text{HNR} = 10 \times \log_{10} \frac{r'_x(\tau_{max})}{1 - r'_x(\tau_{max})} \tag{4.13}$$

4.3 State of the Art Feature Set Generated by Brute Force

It can be seen from Table 4.1 that different researchers have used a variety of features for developing their systems. Some of the features are problem dependent while others are not. Based upon this diversity, a new idea of using brute force for generating features has emerged. In this method, the main goal is to calculate as many features as possible from a speech sample and then use domain knowledge or some feature selection method for reducing the total number of features. Batliner *et al.* (2006) were the first to pool together a large number of features from sets independently developed at different sites for emotion recognition. They combined these feature sets and presented a super set which contains well over 6000 features. In their tests, they did find an improvement by using such a large feature set. Subsequently, Eyben *et al.* (2009) have described an open-source toolkit called OpenEAR for extracting these large number of features (more than 6000 in number).

We calculate a set of 7956 acoustic features from each speech sample using OpenEAR toolkit (Eyben *et al.*, 2009). A total of 68 low level descriptors (LLD) are calculated by dividing the sample into multiple frames of equal length. Delta and double delta functions are calculated for each LLD and 39 statistical functionals are calculated from each LLD and their delta and delta delta functions which make a total of $(68 + 68 + 68) \times 39 = 7956$ features for each speech sample. The details of the LLDs and statistical functions are given in Table 4.2(a) and Table 4.2(b), respectively. For the details about the specific implementation of each LLD, readers are directed to the toolkit's documentation given in Eyben *et al.* (2009).

Table 4.2: Description of 68 low level descriptors and 39 statistical functionals derived using OpenEAR Toolkit (Eyben *et al.*, 2009).

(a) Description of a set of 68 low level descriptors (LLDs)

Feature Groups	**Features in the Group** (68)
Pitch	Pitch (f_o) in Hz and its smoothed contour
Energy	Log Energy per frame
	Energy in frequency bands 0 – 250
	Energy in frequency bands 0 – 650
	Energy in frequency bands 250 – 650
	Energy in frequency bands 1000 – 4000
	Energy in 26 mel-frequency bands
Zerro-crossing rate	Number of zeros crossings and mean ZCR
Cepstrum	13 Mel-frequency cepstrum coefficients
Formants	First three formants and their corresponding bandwidths
Spectral	Centroid, flux, position of spectral max. and min. peaks, spectral roll of points 90%, 75%, 50% and 25%
Voice Quality	Jitter and shimmer, harmonics to noise ratio, probability of voicing

(b) Description of 39 statistical functionals derived from each LLD.

Functionals (39)	**Number**
Relative positions of max./min values	2
Range (max − min)	1
Arithmetic and quadratic means	2
Quartile and inter-quartile ranges	6
5 and 85 percentile values	2
Zero crossings and mean crossing rate	2
Number of peaks and mean distance between peaks	2
Arithmetic mean of peaks	1
Overall arithmetic mean	1
Linear regression coefficients and corresponding approx. error	4
Quadratic regression coefficients and corresponding approx. error	5
Centroid of contour	1
Standard deviation, variance, kurtosis, skewness	4
Arithmetic, quadratic and absolute means	3
Arithmetic, quadratic and absolute means of non-zero values	3

4.4 Summary

In this chapter, we have tried to clarify the different terminologies used in emotion recognition regarding base unit for feature extraction. Then the description of different types of feature sets used and their extraction methods are given. We have also shown how different emotions affect these features for the selected databases given in Section 3.4. Finally, we have described a state of the art feature set consisting of 7956 acoustic features extracted using the OpenEAR Toolkit. As this is a very

large number of features, in the next chapter we examine different feature selection and reduction strategies. Our proposed feature ranking method is also discussed in the next chapter.

Chapter 5

Feature Selection Methods and Results

Intuitively, as the number of features increases, we have more information about the test samples and, hence, we should get better classification accuracy. This simple argument means that classification accuracy should increase monotonically with the number of features for each test sample. However, this is not always observed in the real world data, where increasing the number of features can actually result in the reduction of accuracy. This effect is known as the *curse of dimensionality* (Bellman, 1961).

There are several factors contributing towards this *curse of dimensionality*. Sparsely populated feature spaces, especially when we have a large number of features in comparison to the available training data, noise in the extracted features and irrelevant or redundant features are some of the contributing factors. If we consider the samples as points in a feature space, as we increase the number of features, we are creating a high dimensional space which is sparsely populated by the data. It is easy to see that this will lead to problems in modelling when we have a limited training data.

The inclusion of irrelevant, redundant and highly correlated features can also adversely affect the classification accuracy. Most of the classification algorithms are adversely affected by irrelevant and redundant features. Another issue of having very large number of features per test sample is an increase in the computational time for training the classifiers.

There are many examples of SER systems in the literature where there is no feature selection step. People use the best features reported by other researchers or found in their previous work, which is effectively a manual feature selection method. However, there are algorithms available that can automatically select the best performing features out of a large set.

To alleviate the *curse of dimensionality*, the main goal is to select a subset of d features out of D features, such that $d \leq D$, without significantly degrading the accuracy of the recogniser—or possibly even improving the results. There are two main methods for dimensionality reduction: by transforming the data into another domain which has low dimensions and the second approach is to select the relevant or remove the irrelevant features from the dataset.

We evaluate each feature set described in Chapter 4 independently as well as combined together on all of the selected databases. Then we compare the results of applying traditional feature selection and reduction methods along with our proposed feature ranking methods based upon preferential voting. In the end we analyse the selected features and propose a method to select a 'universal feature set' that can work reasonably well on any emotional speech database.

5.1 Dimensionality Reduction by Domain Transformation

One of the methods for reducing the dimensions of the input training dataset (x_i, y_i), where $x_i \in \mathbb{R}^D$ are the training samples and $y_i \in \{-1, 1\}$ are the class labels, is to project x_i into a low dimensional space such that projected data $\hat{x}_i \in \mathbb{R}^d$ where $d \leq D$. There are several methods for doing this projection. One of the most commonly used is principal component analysis (PCA). It exploits the relevance of features among each other and projects the data onto orthogonally separated dimensions.

This transformation is defined in such a way that the first principal component is in the direction of the highest variance of the data, and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceeding components. In this way, the variance of the data can be captured by using only the first d-principal components where $d < D$. PCA is used for dimensionality reduction independent of the classifier. Other data projection methods that can be used are linear discriminant analysis and singular value decomposition. In this thesis we have used PCA for dimensionality reduction.

5.2 Dimensionality Reduction by Feature Selection

Dimensionality of the input data can also be reduced by selecting d relevant features, from a total of D features, that give good separation of the classes. This task is not very easy as out of total 2^D possible options somehow we have to select one combination which suits our problem. For a fairly large value of D, this task of testing all possible combinations and selecting one that performs the best becomes virtually impossible even for modern day computers. For this reason, some suboptimal feature selection methods

are used that do not give the globally best set of d features but select those features that work reasonably well.

In general, any feature selection algorithm has two parts: a search engine which searches for candidate features and an evaluation function, which tests the candidate features for their fitness. Based upon the nature of evaluation function, current feature selection algorithms can be categorised into two different frameworks: *wrappers* and *filters* (Van der Maaten *et al.*, 2009). In the wrapper framework, the evaluation function incorporates the classification algorithm itself for evaluation and selection of candidate features. In the filter framework, features are evaluated on a performance metric based entirely on the properties of the training data, without the inclusion of a classification algorithm which will eventually use the selected features. Wrapper based feature selection methods are slow and computationally very expensive but are optimised for a specific dataset and classification algorithm. On the other hand, filter based methods are not best optimised for a particular classifier but can be very fast as compared to wrapper based methods while performing reasonably well.

A compromise between the computational cost and optimisation to the dataset and classifier can be achieved by using hybrid methods. These strategies use the wrapper framework as the search engine while the filter framework is used as an evaluation criterion. In the next few sections, we give the details of the feature selection methods that have been used in this thesis.

5.2.1 Wrapper Based Feature Selection Methods

One of the commonly used wrapper based feature selection methods is sequential forward selection (SFS) which is based on a greedy search strategy under the monotonic assumption. The selected features are evaluated on their classification capabilities by using a classifier as an evaluation criterion. The search strategy starts with a single best performing feature, then the next best performing feature in combination with the already selected feature(s) is added to the selected feature set. Only those features are considered that have not been previously selected on the basis of the evaluation criterion.

Given a feature set consisting of D features, $\boldsymbol{F} = \{f_i | i = 1 \dots D\}$, let $J(\theta)$ be our evaluation criterion. The aim of SFS is to select a subset $S^d \subset \boldsymbol{F}$ where $d < D$, starting from empty set $S^0 = \{\}$. A feature f_i will be selected based upon its effect on the evaluation criterion $f_i = \underset{f_i \notin S^d}{\operatorname{argmax}} J(S^d + f_i)$. The process continues sequentially by selecting features as long as the resulting accuracy is increasing monotonically. The process terminates as soon as the recognition rate drops after adding a new feature. The pseudo code for SFS is given in Algorithm 5.1.

Since the selection strategy is greedy, the selected feature set is not in general optimal, i.e., SFS does not necessarily find the best possible combination of features that achieves

Algorithm 5.1 Pseudo Code for Sequential Forwarding Selection (SFS)

Input: $\boldsymbol{F} = \{f_i | i = 1 \dots D\}$
Output: $\boldsymbol{S}^d$ where $d < D$
1: $S^0 \leftarrow \{\}$
2: oldcriterion $\leftarrow 0$
3: newcriterion $\leftarrow 1$
4: **while** newcriterion $>$ oldcriterion **do**
5: oldcriterion $\leftarrow J(S^d)$
6: $f^+ := \underset{f_i \notin S}{\operatorname{argmax}} J(S^d + f_i)$
7: $S^{d+1} \leftarrow \{S^d \cup f^+\}$
8: newcriterion $\leftarrow J(S^{d+1})$
9: $d \leftarrow d + 1$
10: **end while**

best accuracy since it only considers the performance of individual features in combination with currently selected features. This algorithm is very slow, as on every step it has to search for all the combinations of previously not selected features with the currently selected feature set. One of the ways to speed up the process is to select more than one best performing feature in step 6 of Algorithm 5.1.

Sequential backward elimination (SBE) is similar to SFS except that the search strategy for feature selection works in the *backward* direction. Instead of starting with a single best performing feature like SFS, the search strategy for SBE starts with all features included in the selected feature set and works backwards by excluding the worst performing features from the selected feature set. Again a classifier is used for evaluating the performance of the selected features. The process continues as long as the resulting accuracy is increasing monotonically and it stops as soon as the accuracy drops after adding any new feature.

There are several variations of SFS and SBE, one is the "plus l/take away r algorithm" in which SFS is applied l times followed by SBE applied r times. In this case, the method allows a fixed step backtracking defined by the values of l and r, thus previously selected features can be excluded by the backtracking step. There is a variation of this algorithm called sequential floating forward selection (SFFS) (Pudil *et al.*, 1994). This consists of applying several backward elimination steps after each SFS step as long as the resulting subsets perform better than the previously evaluated ones at that level. Thus backtracking in this algorithm is controlled dynamically. Pseudo code for the SFFS method for feature selection is given in Algorithm 5.2.

5.2.2 Filter Based Feature Selection Methods

In filter based feature selection methods, each feature f_i where $i = 1 \dots D$ is independently evaluated and ranked based upon some performance measure. This evaluation is

Algorithm 5.2 Pseudo Code for Sequential Floating Forwarding Selection (SFFS)

Input: $\boldsymbol{F} = \{f_i | i = 1 \dots D\}$
Output: $\boldsymbol{S}^d$ where $d < D$

$S^0 \leftarrow \{\}$
oldcriterion $\leftarrow 0$
newcriterion $\leftarrow 1$
Step 1: (*Inclusion*)
$f^+ := \underset{f_i \notin S}{\text{argmax}}\, J(S^d + f_i)$
$S_{d+1} = S_d + f^+$
oldcriterion $\leftarrow J(S^{d+1})$
Step 2: (*Conditional Exclusion*)
$f^- := \underset{f_i \in S^d}{\text{argmax}}\, J(S^d - f_i)$
newcriterion $\leftarrow J(S^d - f^-)$
if newcriterion $>$ oldcriterion **then**
    $S^{d-1} := S^d - f^-$
    go to Step 2: (*Conditional Exclusion*)
else
    go to Step 1: (*Inclusion*)
end if

performed on the training data only. Based upon the performance of each feature, it is assigned a rank which varies from 1 to D, i.e., 1 is the best performing feature while D is the worst performing feature. After ranking, the first d best performing features are selected.

In this section we will discuss a few filter based feature selection methods that have been used in this work. The problem of using these ranking methods gets more complex considering that each performs differently depending on the dataset and evaluation criterion. Therefore, one method performing well on some data is not guaranteed to work similarly well on another dataset. Similarly, two feature ranking methods may rank the same feature differently based upon the corresponding evaluation criterion. One solution to overcome this problem is to use many filter based ranking methods and then combine their results by some form of feature rank fusion. In the next few sections, we give some basics about the filter based feature selection methods that we have used. Then we will explain our method for feature rank fusion which is based upon a preferential voting scheme used in some forms of elections.

5.2.2.1 Correlation Based Feature Selection

The correlation based feature selection (CFS) method was proposed by Hall (1999) in which features are ranked based upon the following hypothesis: "*that good feature sets contain features that are highly correlated with the class, yet uncorrelated with each other*". From this hypothesis we define two terms: feature-class correlation (r_{yf_i}),

i.e., how well the feature f_i is correlated to the class labels y, and feature-feature correlation ($r_{f_i f_j}$), which is the inter-feature correlation between two features f_i and f_j. If Rc is the feature ranking method, then the rank of feature set S consisting of k features with $k < D$ is given by:

$$Rc_S(k) = \frac{k\overline{r_{yf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{5.1}$$

where $\overline{r_{yf}}$ is the average feature-class correlation, and $\overline{r_{ff}}$ is the average feature-feature correlation as given below:

$$\overline{r_{yf}} = \frac{r_{yf_1} + r_{yf_2} + \ldots r_{yf_k}}{k}$$

$$\overline{r_{ff}} = \frac{r_{f_1 f_2} + r_{f_1 f_3} + \ldots r_{f_k f_1}}{\frac{k(k-1)}{2}}$$

However, in our case we are interested in ranking individual features i.e., $k = 1$. Hence, Equation 5.1 becomes:

$$Rc(f_i) = r_{yf_i} \tag{5.2}$$

Hence, the rank of a feature f_i based upon correlation criterion is equal to its correlation with the class labels y. After the ranking process, we select the top ranked d features.

5.2.2.2 Mutual Information Based Feature Selection

Peng *et al.* (2005) proposed a feature selection scheme which is based upon mutual information and is referred to as maximum relevance and minimum redundancy (mRMR). In terms of maximum relevance, the purpose of feature selection is to find a feature set S with k features which jointly have the largest mutual information with the target class labels y. It has the following form:

$$\max D(S, y) : \; D = \frac{1}{|S|} \sum_{f_i \in S} I(f_i, y) \tag{5.3}$$

where I is the mutual information. The mutual information between two random variable x and y is mathematically defined in terms of their probability density functions as:

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy \tag{5.4}$$

The minimum redundancy condition between two features f_i and f_j is given by:

$$\min R(S) : \; R = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i, f_j) \tag{5.5}$$

The maximum relevance and minimum redundancy (mRMR) criterion is obtained by combining the two constraints:

$$\max_S F(D,R),\ F = \max_S [D-R] \tag{5.6}$$

$$= \max_S \left[\frac{1}{|S|} \sum_{f_i \in S} I(f_i, y) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i, f_j) \right] \tag{5.7}$$

Let us denote $I(f_i, y)$ and $I(f_i, f_j)$ by constants r_{yf_i} and $r_{f_i f_j}$, respectively. We wish to rank that feature highest which is most correlated to y and least correlated to all other features. Therefore, the ranking criterion for a feature (f_i) becomes:

$$Rc(f_i) = r_{yf_i} - \frac{\sum_{j=1, i \neq j}^{D} r_{f_i f_j}}{D} \tag{5.8}$$

Hence, a feature is ranked based upon its relevance with the class labels and its redundancy with other features.

5.2.3 Hybrid Feature Selection Based on SVM Weights

There are several examples in the literature where SVMs have been used for feature selection. In some cases they have been used in wrapper mode while in others in filter mode. Here, we look at two algorithms that use the weights of the support vectors for ranking the quality of selected features. The algorithm uses backward elimination for feature search while using the support vector weights calculated by SVM algorithm as the evaluation criterion for feature elimination. This is why we have put this method separate from the others and categorise it as a hybrid method.

The evaluation criterion for this method is based upon the normal vector w obtained from the linear SVM. As seen before in Section 3.1, the linear SVM has an output of the form

$$y = \text{sign}(\langle w, x \rangle + b) \tag{5.9}$$

where w is normal vector and y is the output class label. The feature selection scheme is based upon the idea that a feature with weight close to zero will have a small effect on the classification accuracy and hence can be removed. The removed feature is the one which has minimal variation on $\|w\|^2$. The ranking criterion Rc for the ith feature is given as:

$$Rc(f_i) = \left| \|w\|^2 - \|w^{(i)}\|^2 \right| \tag{5.10}$$

$$= \frac{1}{2} \left| \sum_{k,j} \alpha_k \alpha_j y_k y_j \boldsymbol{K}(x_k, x_j) - \sum_{k,j} \alpha_k^{(i)} \alpha_j^{(i)} y_k y_j \boldsymbol{K}^{(i)}(x_k, x_j) \right|$$

where $\boldsymbol{K}^{(i)}$ is the kernel matrix of the training data when the ith feature is removed and $\alpha_k^{(i)}$ is the corresponding solution to the SVM quadratic optimisation problem given by Equation 3.13 and $w^{(i)}$ is the corresponding normal vector. For the sake of simplicity and computational cost of the algorithm, $\alpha_k^{(i)}$ is considered equal to α_k even after the training sample i has been removed. For the feature search strategy, the algorithm uses backward elimination. It starts with all features in the selected feature set and removes one feature at a time until d features are left. In the case where D is very large, more than one feature can be removed at each step. This approach has been proposed by several authors like Rakotomamonjy (2003) and Mladenić and Brank (2004).

Two approaches can be used for ranking the features based upon this criterion:

- Zero-order method: In this case, the ranking criterion Rc given in Equation 5.10 is directly used for feature ranking and the method removes the ith feature that produces smallest change in Rc.
- First-order method: In this case, the ranking is done by using the gradient of Equation 5.10. This approach differs from the previous one since features are ranked according to their influence on the absolute value of the derivative of the weight vector, i.e., $Rc\,(k) = \left|\bigtriangledown(\|\boldsymbol{w}\|^2 - \|\boldsymbol{w^{(i)}}\|^2)\right|$.

In this thesis we do not give the mathematical formulation for this equation. We only use this formulation for feature selection. However, interested readers are directed to the paper by Rakotomamonjy (2003) for mathematical details of the formula.

5.3 Feature Rank Fusion Using Preferential Borda Voting

In the previous few sections, we have explained different wrapper, filter and hybrid feature selection methods. The output of each filter method is a rank of each individual feature, i.e., how suited that feature is for separating the target classes, and the rank value of each feature. The problem of using these ranking methods gets more complex considering that each performs differently even on the same dataset. One solution to overcome this problem is to combine the results of several different feature selection methods and find the overall best performing feature set.

The main idea is that a good feature will be generally ranked high by all rankers. A feature that gets a high rank only by one ranker is most likely not the overall best performing feature. What we want to achieve is to remove or minimise the effect of such a random ranked feature while enhancing the rank of the features that are ranked high by all or most of the rankers.

For feature rank fusion, the majority voting scheme seems to be the obvious candidate. However, majority voting can only be applied when we have a large number of voters

as compared to the candidates. In the feature ranking case, the scenario is reversed; we have more candidate features than the number of voters (rankers). Hence one can not directly apply the majority voting scheme.

To solve this problem, we propose to use a preferential voting scheme called 'Borda voting' (de Borda, 1781). This scheme is named after 18th century French mathematician Jean-Charles de Borda. The main difference between this voting method and other standard elections is that the winner of the elections is not the candidate who gets maximum votes, rather the one who is most preferred by the general population.

Let us consider an example of a special election for the president of a country. Here, *voters* are the general public who express their opinion by casting their votes. When casting a vote, a voter chooses a rank order for all the candidates in which they prefer them to be the president. The Borda voting method then computes the mean rank of each candidate over all voters. The top ranked candidate is declared the winner of the elections. So the candidate who is most preferred by the voters, in general, wins the elections.

In our case, the number of candidates, i.e. features, is very large as compared to the number of voters, i.e., feature ranking methods (L), where $L << D$. Using Borda voting, individually ranked lists of features are combined to produce a new ranking. By combining the results of several different feature ranking methods, the feature which generally gets good ranks by all of the ranking methods will get a high rank in the combined feature set. By the same argument, the random performing feature will get the lower rank. Thus we should be able to separate the good performing features from the not so well performing features. Secondly by using several rankers, we are reducing the effect of a random ranker as well.

The output of each of the L rankers is considered as a vote $\{V_1, V_2, \dots V_L\}$ and these are fused together to get the final rank of features. Intuitively, as L, becomes large, the influence of a random voter, i.e., bad feature ranker, decreases. For a feature set $\boldsymbol{F} = [f_1, f_2, \dots, f_D]$ consisting of D features, the fusion function can be defined as:

$$\boldsymbol{F}' = \Gamma\{Rc_i\,(\boldsymbol{F})\} \tag{5.11}$$

where $Rc_i = [Rc_1, Rc_2, \dots, Rc_L]$ are the L feature rankers, Γ is the fusion function and $\boldsymbol{F}'$ are the new ranks or scores of the features. There are two possible ways to combine these ranks: using scores of each feature (soft decision) or using the rank of each feature (hard decision).

5.3.1 Soft Decision Feature Fusion

Let $0 \leq Rc_i(f_j) \leq 1$ and $r_{i,j}$ be the normalised ranking score and rank of feature j when using feature ranking method Rc_i respectively. In this approach, soft output of the ranker, i.e., feature ranking score $Rc_i(f_j)$ is used for the fusion process which is given by:

$$f'_j = \text{sort}\left\{\Gamma_{i=1}^{L} Rc_i(f_j)\right\},\ 1 \leq j \leq D,\ f_j \in \mathbf{F} \tag{5.12}$$

where Γ can be any function like $\min, \max$, sum, product or mean over variance. In our feature selection test, we have used the summation function. The first d features out of the newly ranked feature set (f'_i), are the best overall ranked features.

5.3.2 Hard Decision Feature Fusion

In hard decision Borda ranking, the rank $r_{i,j}$ of each feature f_i is used for getting the final rank of each feature:

$$f'_j = \text{sort}\left\{\Gamma_{i=1}^{L} (D - r_{i,j})\right\},\ 1 \leq j \leq D \tag{5.13}$$

From now, we are going to refer to these methods as Borda-Soft and Borda-Hard decision ranking, respectively.

5.4 Results Using Full Feature Set

In this section we will see the effect of using each feature group mentioned in Section 4.3 on classification accuracy independently as well as combined together for three acted emotional speech databases and the two separate parts of the spontaneous emotional speech database.

5.4.1 Results on Three Acted Emotional Speech Databases

Table 5.1 shows percentage UA and WA accuracies and the corresponding standard deviations for speaker dependent cross validation (SD–CV) for the three acted emotional speech databases namely, DES, Berlin and Serbian. For classification, we have used each feature set independently as well as combined together with linear SVM classifier with $C = 0.1$ in one-versus-one configuration.

The test samples in the DES database are balanced for each emotional class, therefore, the UA and WA accuracies are the same. Out of the prosodic feature group (pitch, energy, ZCR), energy features give the best results. Pitch information does not seem to

Table 5.1: Classification accuracy with standard deviation on the selected three acted emotional speech databases using linear SVM.

Feature Group	**DES**		**Berlin**		**Serbian**	
	UA (%)	**WA (%)**	**UA (%)**	**WA (%)**	**UA (%)**	**WA (%)**
Pitch	13.2 (4.8)	13.2 (4.8)	47.1 (3.1)	51.6 (1.8)	46.7 (2.9)	46.7 (2.9)
Energy	**61.9** (5.8)	**61.9** (5.8)	**82.2** (1.5)	**83.4** (2.0)	89.5 (1.2)	89.5 (1.2)
ZCR	27.3 (2.6)	27.3 (2.6)	53.5 (5.8)	56.5 (6.1)	52.3 (2.3)	52.3 (2.3)
MFCC	52.4 (4.4)	52.4 (4.4)	84.0 (4.2)	84.5 (3.5)	**90.9** (1.2)	**90.9** (1.2)
Formant	35.0 (6.9)	35.0 (6.9)	54.1 (7.9)	55.1 (7.7)	58.0 (1.2)	58.0 (1.2)
Spectral	49.8 (4.4)	49.8 (4.4)	66.7 (5.9)	68.0 (5.8)	81.2 (1.2)	81.2 (1.2)
Voice Quality	26.5 (9.8)	26.5 (9.8)	50.8 (3.0)	54.0 (2.6)	50.1 (1.0)	50.1 (1.0)
SD–CV(all)	68.1 (7.3)	68.1 (7.3)	87.2 (4.1)	87.9 (3.4)	93.5 (1.5)	93.5 (1.5)
SI–CV(all)	47.7 (8.4)	47.7 (8.4)	74.9 (10.6)	79.2 (8.2)	78.6 (5.5)	78.6 (5.5)

give any information about the classes in this database as the average percentage results are even below the chance level which is 20% for this database. The next best performing features are the MFCCs. As mentioned earlier, these features are used in ASR systems in which the main goal of the recogniser is to identify 'what' is being spoken. However, the results show that the MFCC features can give reliable information about 'how' something is being spoken. The best results are obtained by using all of the extracted features combined together shown as SD–CV(all). Table 5.1 also shows SI–CV results that are much lower than the SD–CV results meaning that it is much harder to classify emotions when the classifier has absolutely no information about the current speaker. In such situations, there are few established methods that can be used to adapt the trained model for the unknown speakers which we shall explore in Chapter 7.

In the case of the Berlin emotional speech database, the unweighted chance level accuracy is around 14%. Out of the three prosodic features, energy features perform the best. However, pitch and ZCR feature sets perform much better than the chance level which was not the case for DES database. Out of the spectral features, MFCCs are by far the best performing feature subset achieving very high accuracy just on their own. The combination of all feature sets gives the best performance, although many feature sets that were not performing very well individually perform well in combination with each other.

For the Serbian database, all of the individual feature sets give better classification accuracy than the 20% chance level accuracy. Out of the prosodic features, energy features perform a lot better than the other two. MFCC features again perform very well out of the spectral features while pitch and voice quality are the two worst performing feature sets. The combined feature set again performs the best indicating that uncorrelated features, which may not be performing very well on their own, can perform well in combination to other features.

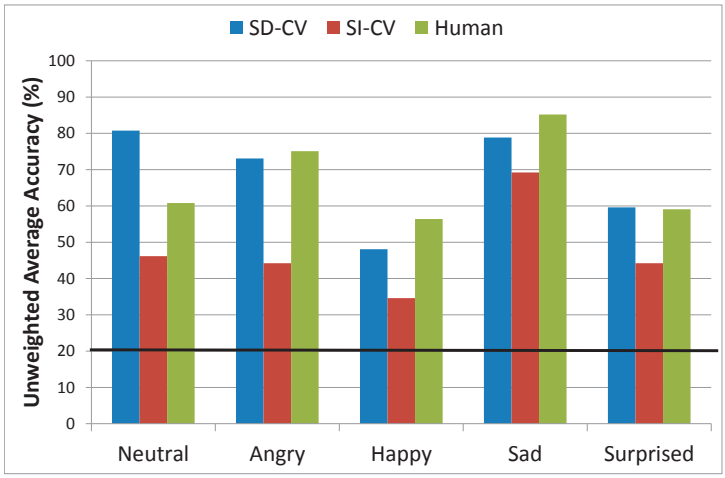
Generally looking at the results, DES is the most difficult database out of the selected three. For this database, the reported human accuracy is also the lowest. One of the reasons for this low recognition rate could be the quality of the recordings or the general traits of Danish language. Out of the feature sets, energy and MFCC features are performing the best while pitch and voice quality features are performing the worst. These results are not in exact agreement with Murray and Arnott (1993) who found prosodic and voice quality features to be the most effective feature sets for emotion recognition. However, these results were based upon psychological studies and were used for emotion synthesis. This does not mean that the worst performing features do not give any information about the underlying classes. Usually adding the 'not well' performing features in the selected feature set should decrease the performance, but features that give best classification performance are not always the best performing individual features. Rather they are the ones that perform best in combination with other features. This is observed in Table 5.1 in which the combined feature set, which contains all of the individually best and worst performing features, gives the best classification performance for all of the three databases.

5.4.1.1 Comparison With Human Accuracy

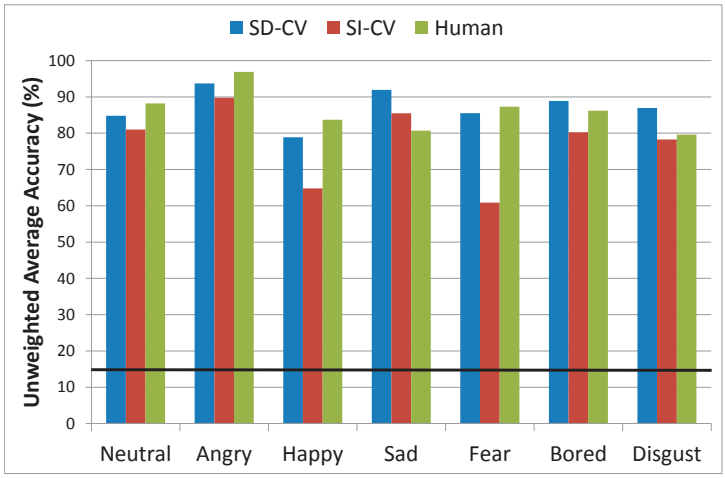
Figure 5.1(a) shows the per class UA accuracies for DES database for SD–CV and SI–CV using all features in comparison with human accuracies reported in Engberg and Hansen (1996). It is clear from the figure that SI–CV results are much lower than the SD–CV meaning that it is much harder to classify emotions when the classifier has absolutely no information about the current speaker to adapt its models. An important thing to notice is that the easiest class to separate is *neutral* for which the classifier exceeds human accuracy for SD–CV. The most difficult emotional classes to identify are *happy* and *surprised* which is consistent with the human labellers who also found these two classes most difficult to identify.

Figure 5.1(b) shows the SD–CV and SI–CV accuracies for Berlin database in comparison to human accuracy reported in Burkhardt *et al.* (2005). *Happy* and *fear* emotions have worst performance for both SD–CVand SI–CV. Whereas for *disgust* and *sad* emotions, the machine surpasses the performance of human labellers. One of the reasons for this could be that human listeners were made to label the utterance by listening to it only once and in random order. Humans usually take some time to adapt to the speaking style. This could be one of the reasons for poor performance especially on these two emotions.

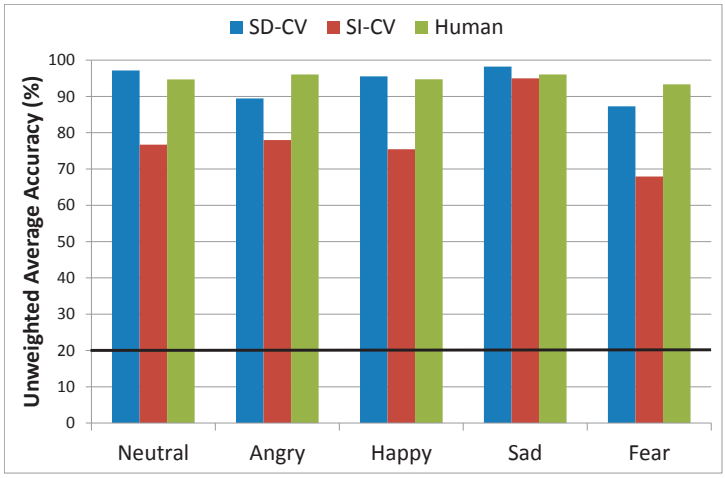
Figure 5.1(c) shows the comparison of SD–CV and SI–CV accuracies for the Serbian database with human accuracies reported in Jovicic *et al.* (2004). Using linear SVMs, best classification accuracy is obtained for the *neutral* emotion while worst results are for *fear* and *angry*. Both of these emotions have negative *valence* and high *arousal*.



(a) DES



(b) Berlin



(c) Serbian

Figure 5.1: SD–CV and SI–CV percentage UA classification accuracies in comparison with reported human accuracies using all features and SVM classifier for (a) five-class DES database; (b) seven-class Berlin database and (c) five-class Serbian database. The horizontal black line is the UA chance level accuracy for each corresponding database.

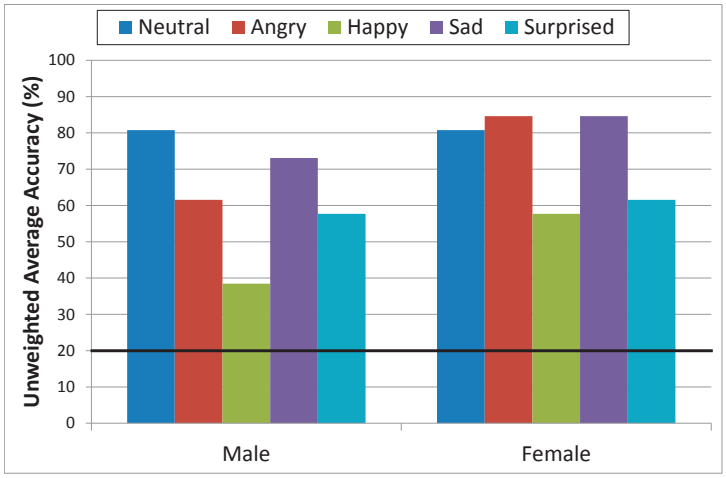
In the reported human perception tests for the Serbian database, human labellers managed much higher performance on these two emotion classes. It has been established that humans take some time to adapt to the speaking style of the speaker. This is why we find it easier to understand the emotions of the people whom we know quite well as compared to unknown people. For this particular database, 30 student labellers were allowed to listen to all of the speech samples from one speaker and then label the target speech samples. They were allowed to listen to the samples as many times as they wished thus giving them time to adapt to the particular speaker. This is why the human accuracy for these two emotions *angry* and *fear* is very high.

5.4.1.2 Comparison Between Gender

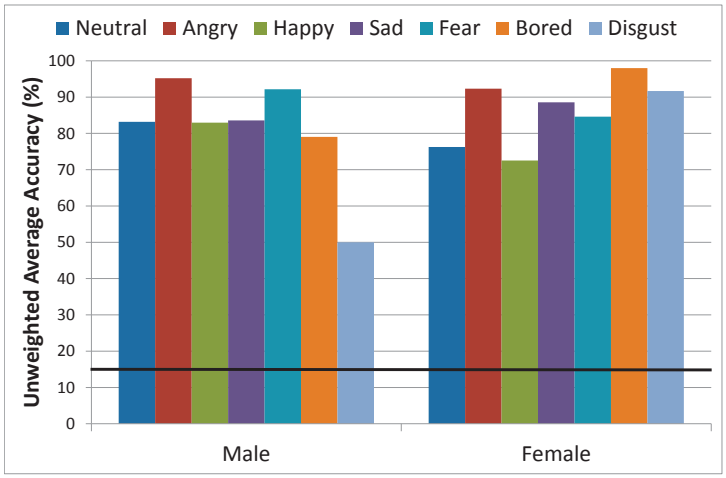
As mentioned earlier, usually SER systems do not consider gender differences as important. We think that these are significant and should be considered in any gender independent SER system. In this section we show these differences in the gender specific results for each database. Figure 5.2(a), Figure 5.2(b) and Figure 5.2(c) show the corresponding accuracies with respect to gender for all of the three databases. For the DES database, classification accuracies for female speakers are much higher than for the male speakers. Major differences in the accuracy can be observed for *anger*, *happy* and *sad* emotion classes for which recognition accuracy for female speakers is much higher than the male speakers. These results indicate that the gender does play quite a significant role in the colouring of emotional speech. Hence the gender independent SER system must consider and compensate for these differences.

For the Berlin database, the distribution of accuracies of emotions among the two genders of speakers are shown in Figure 5.2(b). The overall difference in accuracies between the two genders is not very large as compared to DES, other than for *bored* and *disgust* emotions. Classification performance for *disgust* is much better in female speakers while performance on *bored* is much better for male speakers. This could be because of the quality of actors who portrayed the emotions or the intrinsic nature of male and female speakers to generally express the corresponding emotions. Another important thing to observe is that the average age for male and female speakers in the Berlin database is around 29 and 30 years respectively whereas for DES it is 45 and 44 years respectively. One can infer that as people grow older, their method of portraying speech and emotions changes which is perhaps why we observe a big difference between the accuracies of two genders in DES and not in Berlin Database.

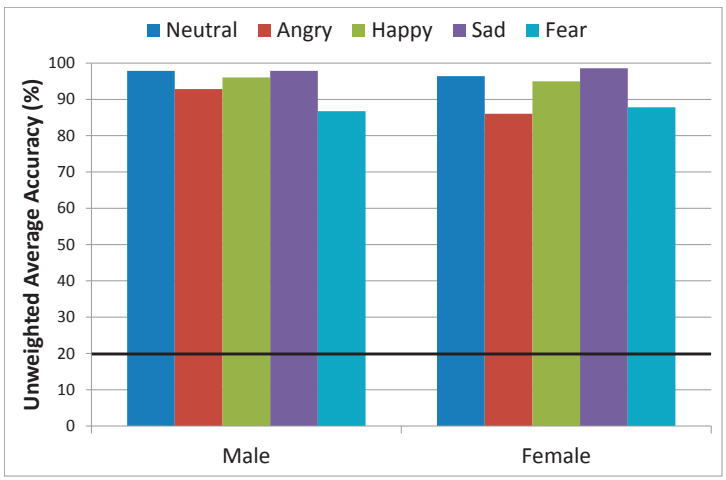
For the Serbian database, all of the emotions have similar accuracy across the gender other than *angry* for which females have much lower accuracy than men. These results are in contradiction with that on DES database where *angry* emotions in female speakers was much easier to identify. Cultural difference could be one of the reasons for this.



(a) DES



(b) Berlin



(c) Serbian

Figure 5.2: Gender specific percentage UA classification accuracies using SI–CV all features and linear SVM classifier for (a) five-class DES database; (b) seven-class Berlin database and (c) five-class Serbian database. The horizontal black line is the UA chance level accuracy for each corresponding database.

Table 5.2: Classification accuracy with standard deviation on the selected three acted emotional speech database using linear SVM.

Feature Group	Aibo-Mont		Aibo-Ohm	
	UA (%)	WA (%)	UA (%)	WA (%)
Pitch	32.1 (2.1)	33.8 (1.9	28.7 (2.4)	32.8 (2.4)
Energy	**48.2** (5.7)	**47.6** (5.0)	**43.5** (3.3)	**46.0** (4.0)
ZCR	31.2 (1.9)	35.0 (1.9)	27.6 (2.4)	30.2 (3.1)
MFCC	44.6 (7.1)	45.4 (6.6)	41.9 (5.2)	45.2 (5.5)
Formant	35.3 (3.5)	36.2 (4.7)	32.6 (3.1)	35.6 (3.1)
Spectral	36.6 (2.9)	36.8 (2.7)	34.8 (2.5)	37.2 (2.4)
Voice Quality	28.2 (2.4)	29.0 (1.6)	29.3 (4.2)	33.2 (4.3)
SD–CV(all)	48.4 (2.6)	47.1 (2.7)	51.6 (1.7)	51.9 (1.7)
SI–CV(all)	36.7 (8.8)	39.6 (8.1)	41.0 (7.8)	43.2 (8.6)

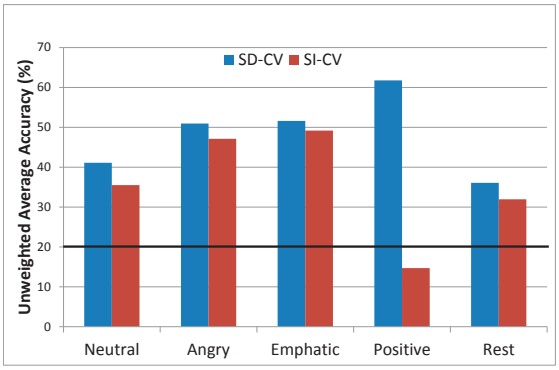
However, there is no information available about the age of the speakers for this database, so we can not infer any further.

5.4.2 Results on Two Spontaneous Emotional Speech Database

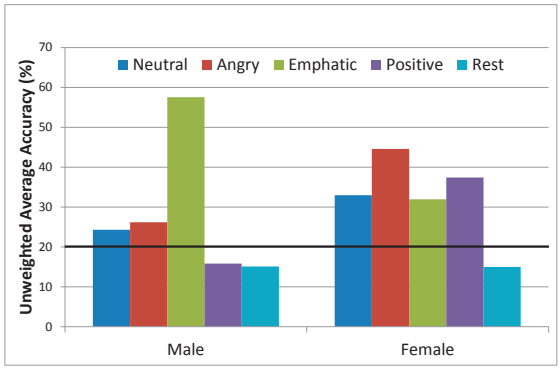
In this section we present the baseline results for each feature subset on the spontaneous emotional speech database FAU Aibo emotion corpus. The details of this database are given in Section 3.4 where we mentioned that it consists of two parts: 'Mont' and 'Ohm', each recorded in a different school. Hence we have decided to handle it as two separate databases. This separation will be very helpful when we apply inter database emotion recognition in Chapter 7.

Table 5.2 shows the classification accuracies on these two spontaneous emotional speech databases. To deal with highly imbalanced data, we apply synthetic minority oversampling (Chawla *et al.*, 2002) to balance the classes during the classifier training phase. From the results we can see that the accuracies on 'Mont' and 'Ohm' for both SD–CV and SI–CV are different, supporting our argument for treating these two databases as separate and independent. For this database, chance level UA is 20% and MFCCs and energy features are the best performing feature sets. Another important observation to be made is that WA is much less than the chance level accuracies which are 65.1% and 56.2% for Aibo-Mont and Aibo-Ohm respectively. As mentioned in Section 3.3.1, for an imbalanced database WA can give a badly biased view of the performance of the classifier.

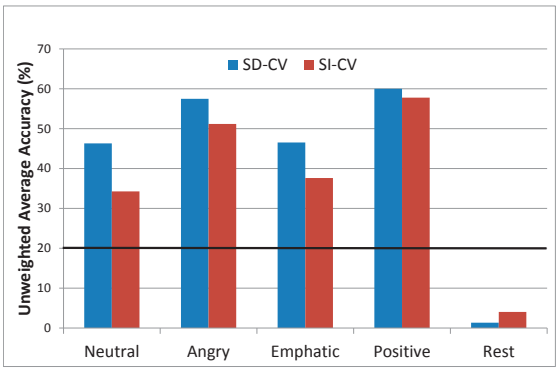
From Figure 5.3(a) and Figure 5.3(c), it is clear that the *rest* class out of the five is the most difficult to recognise. This is quite obvious because this class was designated to those tokens that could not be classified as belonging to any other major class. Best classification accuracy is achieved for *angry* and *positive* emotions.



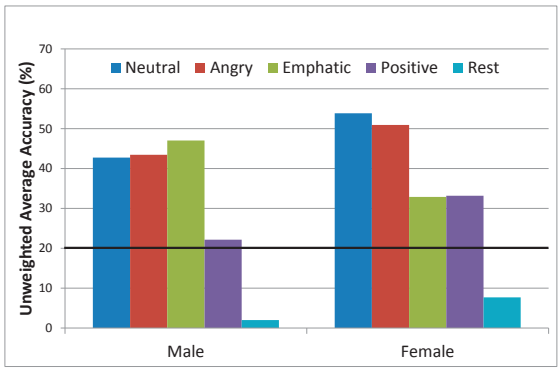
(a) SD–CV and SI–CV % UA for Aibo-Mont



(b) Gender specific results for Aibo-Mont



(c) SD–CV and SI–CV % UA for Aibo-Ohm



(d) Gender specific results for Aibo-Ohm

Figure 5.3: SD–CV and SI–CV percentage UA classification accuracies and gender specific results for Aibo-Mont and Aibo-Ohm database. The horizontal black line is the UA chance level accuracy for each corresponding database.

For gender specific results, UA accuracy for female speakers is higher than for male speakers for Aibo-Mont while this difference is not that significant in the case of Aibo-Ohm. One possible reason is the number of male speakers in Aibo-Mont is much less than the female speakers (8 and 17 respectively). In the case of Aibo-Ohm, the number is equal, i.e., 13 speakers for each gender. Another possible reason could be that on average speakers in Aibo-Mont are one year older than the speakers in Aibo-Ohm who are 10 years old. This difference can make a difference in the development of the vocal system and speaking style. There could be several other reasons for this difference but they are out of the scope of this thesis.

5.5 Results Using Feature Selection

In this section we test the effect of different feature selection methods discussed earlier on the classification accuracy. We will apply the dimensionality reduction method (PCA), wrapper based feature selection (SFS and SBE), SVM based hybrid feature selection methods (SVM-Marg and SVM-MargDiff) and our proposed Borda preferential voting based feature ranking (Borda-Hard and Borda-Soft) along with four other filter based

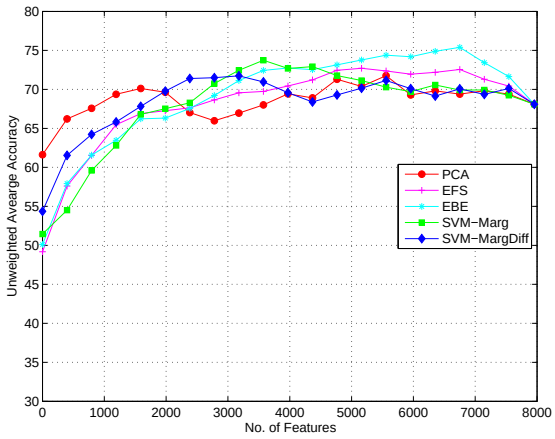
methods. Instead of applying the traditional SFS and SBE in which the feature selection stops as soon as the accuracy drops by adding any new feature, we extend the feature selection/elimination until all the features have been included/excluded from the full feature set. We call these methods extended forward selection (EFS) and extended backward elimination (EBE).

Figure 5.4 shows the SD–CV UA accuracy versus the number of features selected after applying dimensionality reduction, sequential feature selection and ranking methods along with our proposed Borda preferential feature ranking method. For PCA, the horizontal axis shows the number of eigen vectors retained while for EBE, it is the number of features left out from the full feature set.

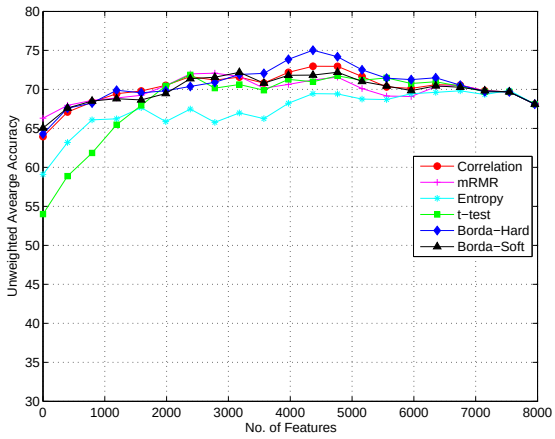
For the DES database, out of the four wrapper methods, EBE and the two SVM based hybrid feature selection methods perform the best as shown in Figure 5.4(a). Peaks of the curves are seen after the algorithm has selected around 2000–4000 features out of 7956 features. In comparison, the feature ranking methods shown in Figure 5.4(b) for the same DES database uses an independent performance measure to evaluate the fitness of a particular feature. These methods are much quicker than wrapper or hybrid feature selection methods. Out of the first four filter methods, correlation and mRMR feature ranking have already been discussed. To increase the number of voters for our proposed hard-decision based Borda ranking (Borda-Hard) and soft-decision based Borda ranking (Borda-Soft), two other standard filter methods have been included.

Results of our proposed Borda-Hard and Borda-Soft feature ranking methods are shown in blue diamonds and black triangles respectively. These results are comparable to computationally expensive sequential feature selection methods and SVM based hybrid methods. Hence, the proposed feature ranking has been able to utilise the rankings of several filter methods and combine them such that the effect of a random ranker is minimised and overall best ranked features get the highest ranks. The effect of adding bad or un-related features is clear from the plots as the classification accuracy decreases as we keep on adding more features.

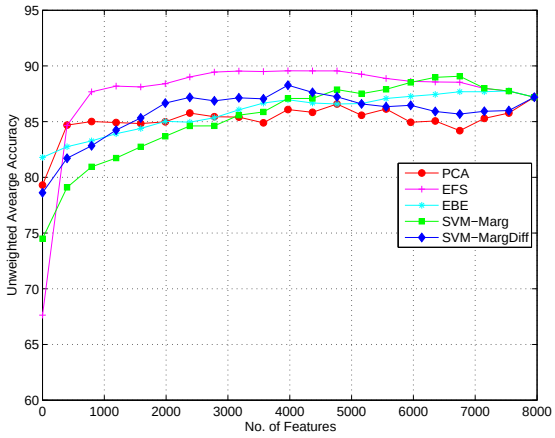
For the Berlin database, out of the wrapper and hybrid methods, EFS performs the best. The second best performing methods is SVM-MargDiff which is a hybrid method. The main reason behind the performance gain for EFS in comparison to SVM-MargDiff is that using this method, only those features are selected that directly increase the classification accuracies while for SVM based hybrid feature selection, features are selected on the basis of their effect on SVM margin. Although there is a potential computational performance gain by using hybrid methods, as the features are not selected on the basis of their direct effect on classification accuracy, it is not guaranteed that the algorithm will select the best features. Out of the many filter ranking based methods used, Borda-Hard performs the best showing the superior capability of our proposed algorithm.



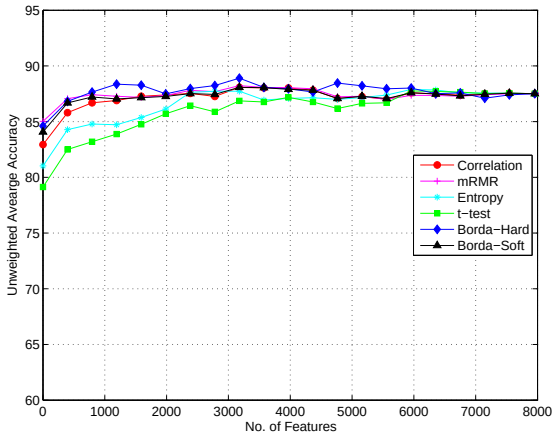
(a) DES



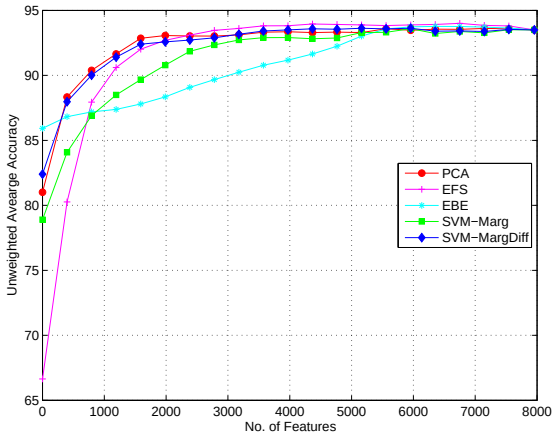
(b) DES



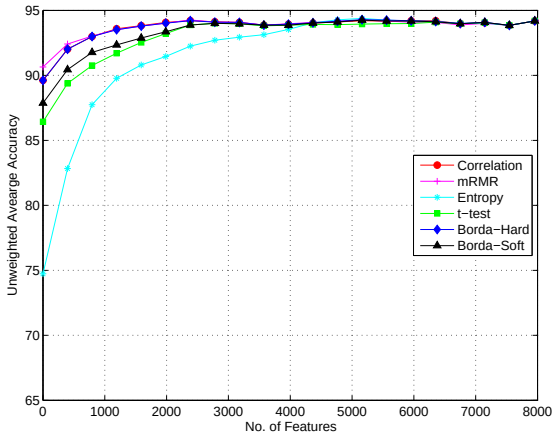
(c) Berlin



(d) Berlin

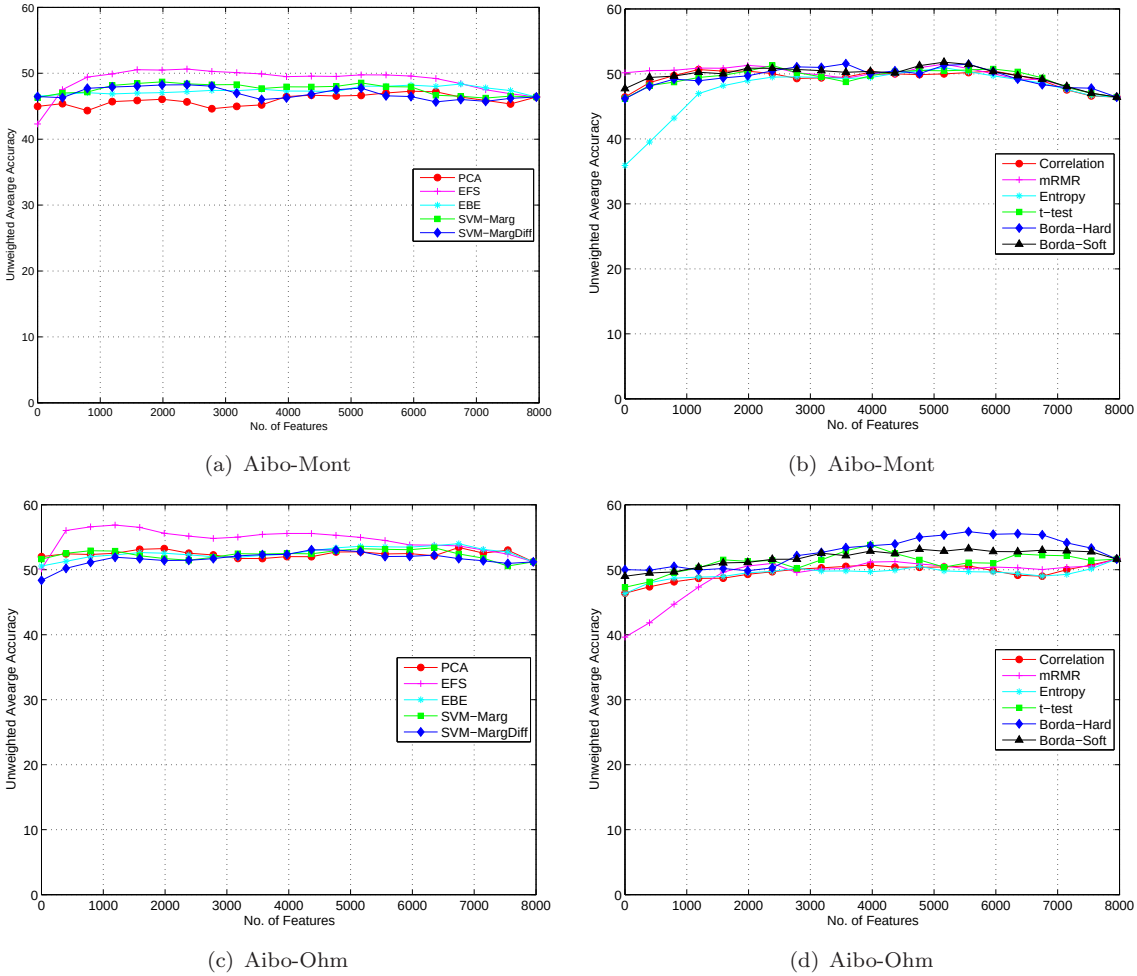


(e) Serbian



(f) Serbian

Figure 5.4: SD–CV percentage UA classification accuracy with respect to the number of features selected for acted emotional speech databases.



(a) Aibo-Mont (b) Aibo-Mont

(c) Aibo-Ohm (d) Aibo-Ohm

Figure 5.5: SD–CV percentage UA classification accuracy with respect to the number of features selected for Aibo-Mont and Aibo-Ohm emotional speech databases.

For the Serbian emotional speech database, out of the wrapper based methods, EFS performs the best closely followed by SVM-MargDiff method. Out of the six filter based methods, Borda-Hard and mRMR based feature ranking methods give similar results.

Figure 5.5 shows the results of applying different feature selection methods for the two spontaneous Aibo-Mont and Aibo-Ohm databases. Again we observe that the proposed preferential feature rank fusion method is able to outperform other ranking methods and its performance is not very far behind the overall best performing EFS wrapper method.

From the plots shown in Figure 5.4 and Figure 5.5, we conclude that out of the wrapper based methods, SFS is the best performing and out of the filter based methods, our proposed Borda ranking based upon hard decision is performing the best. The results of the two algorithms are competitive which means that by combining the results of several feature ranking methods by hard decision Borda ranking, one can strike a balance

Table 5.3: Confusion matrix for 5-class SD–CV emotion classification for the DES database by using linear-SVM classifier with C=0.1 and top 3000 features selected by Hard-decision Borda method.

Input \	Neutral	Angry	Happy	Sad	Surprised
Neutral	**40**	1	0	9	0
Angry	4	**35**	6	1	4
Happy	5	4	**25**	0	16
Sad	6	0	0	**44**	0
Surprised	1	5	12	1	**31**

Table 5.4: Confusion matrix for 7-class SD–CV emotion classification for the Berlin database by using linear-SVM classifier with C=0.1 and top 3000 features selected by Hard-decision Borda method.

Input \	Neutral	Angry	Happy	Sad	Fear	Bored	Disgust
Neutral	**74**	0	1	0	1	3	0
Angry	0	**119**	7	0	0	0	1
Happy	1	11	**54**	0	5	0	0
Sad	1	0	0	**56**	0	5	0
Fear	2	2	3	0	**61**	0	1
Bored	3	0	0	2	0	**75**	1
Disgust	1	1	1	1	3	1	**38**

between accurate but computationally intensive wrapper methods and less accurate but computationally less intensive filter methods.

Analysing Figure 5.4 and Figure 5.5, we conclude that classification accuracy generally increases as we add new features. However, after a certain number of features have been added, it starts to decrease. For our problem, we think that this point comes after 2000–4000 features have been selected by our proposed hard-decision based Borda ranking method. Therefore, we have decided to use the first 3000 features selected by hard-decision Borda ranking for each database.

Table 5.9 shows the results of using only 3000 features selected by Borda-Hard ranking for the selected acted and spontaneous emotional speech databases. The classification results after feature selection are higher than without any feature selection. Tables 5.3–5.7 show the corresponding confusion matrices using SD–CV for all of the five databases. For DES, the two classes most confused with each other are *happy* and *surprised*; for Berlin and Serbian databases, they are *angry* and *happy*. All of these confused classes have high *arousal* but different *valence*. For Aibo databases, worst results are obtained for *emphatic* and *rest* classes which are the two mostly confused with each other. In the next chapter we will look at some ways to visualise the confusion matrix and interpret them more easily.

Table 5.5: Confusion matrix for 5-class SD–CV emotion classification for the Serbian database by using linear-SVM classifier with C=0.1 and top 3000 features selected by Hard-decision Borda method.

Input \	Neutral	Angry	Fear	Happy	Sad
Neutral	**541**	0	2	3	12
Angry	0	**505**	4	49	0
Fear	5	2	**537**	13	1
Happy	3	43	13	**499**	0
Sad	9	0	2	0	**547**

Table 5.6: Confusion matrix for 5-class SD–CV emotion classification for the Aibo-Mont database by using linear-SVM classifier with C=0.1 and top 3000 features selected by Hard-decision Borda method.

Input \	Neutral	Angry	Emphatic	Positive	Rest
Neutral	**4367**	21	93	780	116
Angry	356	**137**	25	40	53
Emphatic	1027	14	**325**	104	38
Positive	4	0	0	**211**	0
Rest	298	0	4	109	**135**

Table 5.7: Confusion matrix for 5-class SD–CV emotion classification for the Aibo-Ohm database by using linear-SVM classifier with C=0.1 and top 3000 features selected by Hard-decision Borda method.

Input \	Neutral	Angry	Emphatic	Positive	Rest
Neutral	**3275**	936	352	209	818
Angry	147	**675**	23	22	14
Emphatic	479	621	**809**	18	166
Positive	103	60	5	**371**	135
Rest	201	149	47	53	**271**

5.6 Search for Universal Features

We saw in Section 5.4 that out of the several different feature sets tested, generally MFCC and energy features perform the best for all of the databases. However, one selected feature set performing well on a certain database may not perform as well on another database. The main motivation for this section is to find a 'universal feature set' which should perform reasonably well on all/any database.

After selecting the best performing features on all of the databases individually, we have one feature set per database which performs very well on its corresponding databases. One of the ways to search for the universal features is to combine all of the individually best performing features together in a one big set consisting of the union of all individual sets. Another way is to apply Borda ranking on the individually selected features to get the top ranked universal features.

Table 5.8: List of Universal features selected by Borda-Hard features selection from all of the five database.

Feature Group	**Universal-Small**	**Universal-Large**
Pitch	72	164
Energy	1029	2528
ZCR	47	83
Cepstrum	1252	1145
Formant	118	386
Spectral	347	720
VQ	149	285
Total	**3000**	**5311**

We apply Borda-Hard ranking on features individually selected from each database being tested. By doing so we expect that the top ranked features from these several databases should be universal. By taking the union of the 3000 feature selected from each database independently, we get a large set consisting of 5311 features (Uni-Large). We apply Borda-hard on these to select the top 3000 universal features (Uni-Small). This number has been selected to allow us to compare our results with the individual feature sets. Table 5.8 shows the number of universal features in each feature group. We can see that out of the prosodic features, energy features have highest representation in the selected universal feature set. Out of the Uni-Small, 57% of the features (Cepstrum and Spectral) are based upon the spectrum. Energy and spectrum based features comprise more than 92% of the selected universal feature set. From this, we can conclude that the most effective feature sets for emotion recognition from any database are energy and spectrum based features.

Table 5.9 shows the results of applying SD–CV and SI–CV on the selected databases using the first 3000 features selected by Borda-Hard feature ranking as well as the two universal feature sets. From the results we can see that feature selection significantly improves the classification accuracy. Out of the two universal feature sets, the Uni-Large features, selected by the union of individual sets, perform better than Uni-Small. Perhaps for capturing the varied emotions across several databases, we need an equally varied feature set. The results for both universal feature sets are competitive.

These results indicate that the feature selection using Borda-Hard ranking methods improves the classification performance significantly. The proposed method for selecting universal feature sets also works reasonably well.

5.7 Summary

In this chapter, we gave some background about feature selection methods. We discussed the wrapper and filter based feature selection methods as well as the hybrid feature

Table 5.9: Unweighted and weighted average percentage accuracies for speaker dependent 10-fold and speaker independent leave one speaker out cross validation without any feature selection, with feature selection using Borda-Hard method and the two universal feature sets.

(a) Speaker-Dependent 10-fold Cross Validation

Database		**All**	**3000**	**Uni-Large**	**Uni-Small**
DES	UA	68.1 (7.3)	69.6 (6.2)	68.8 (7.8)	68.0 (6.7)
	WA	68.1 (7.3)	69.6 (6.2)	68.8 (7.8)	68.0 (6.7)
Berlin	UA	87.2 (4.1)	88.2 (2.6)	87.2 (2.3)	87.0 (3.1)
	WA	87.9 (3.4)	89.2 (2.4)	87.7 (1.8)	87.6 (2.5)
Serbian	UA	93.5 (1.5)	94.2 (0.8)	94.1 (0.4)	93.2 (0.8)
	WA	93.5 (1.5)	94.2 (0.8)	94.1 (0.4)	93.2 (0.8)
Aibo-Mont	UA	48.4 (2.6)	49.9 (2.4)	49.9 (3.3)	49.1 (3.0)
	WA	47.1 (2.6)	48.9 (2.7)	48.9 (2.1)	48.1 (2.4)
Aibo-Ohm	UA	51.6 (1.7)	53.3 (1.7)	52.5 (1.3)	52.1 (1.1)
	WA	51.9 (1.7)	52.8 (1.8)	52.8 (1.4)	52.5 (1.2)

(b) Speaker-Independent Leave One Speaker Out

Database		**All**	**3000**	**Uni-Large**	**Uni-Small**
DES	UA	47.7 (8.4)	49.6 (6.9)	46.9 (7.2)	46.5 (7.2)
	WA	47.7 (8.4)	49.6 (6.9)	46.9 (7.2)	46.5 (7.2)
Berlin	UA	74.9 (10.6)	77.4 (9.6)	76.6 (9.9)	76.2 (9.7)
	WA	79.2 (8.2)	79.9 (8.5)	80.6 (7.7)	80.3 (7.7)
Serbian	UA	78.6 (5.5)	79.0 (5.9)	79.0 (6.2)	78.9 (6.1)
	WA	78.6 (5.5)	79.0 (5.9)	79.0 (6.2)	78.9 (6.1)
Aibo-Mont	UA	36.7 (8.8)	39.4 (7.9)	39.3 (8.8)	39.0 (8.1)
	WA	39.6 (8.1)	43.2 (10.5)	42.4 (12.6)	42.6 (11.8)
Aibo-Ohm	UA	41.0 (7.8)	42.4 (8.5)	41.8 (8.4)	41.8 (8.6)
	WA	43.2 (8.6)	43.5 (9.3)	43.7 (9.1)	43.5 (9.3)

selection methods based upon SVMs used in this thesis. Then we gave the details of a novel feature rank fusion methods based upon Borda preferential voting that can be used to combine the results of several feature ranking methods.

These details are followed by classification results on the selected databases using the complete feature sets and the selected features by the proposed method. After a thorough comparison of several feature selection methods, we conclude that our proposed hard decision based Borda ranking method performs very well. In the end, we present a method to search for 'universal features' across several databases. The selected 'universal feature' sets perform competitively. In the next chapter we discuss and compare several methods for extending binary SVM classification to multiclass classification problems and propose a data driven approach for deriving the hierarchical structure of SVM classifiers.

Chapter 6

Hierarchical Classification and Results

In the previous chapter, all the results presented were by using SVMs in one-versus-one configuration. In this chapter, we compare four ways to extend binary support vector machines (SVMs) to multiclass classification for recognising emotions from speech. Analysis of the errors made by these classifiers leads us to apply non-metric multi-dimensional scaling (NMDS) to produce a compact (two-dimensional) representation of the data suitable for guiding the choice of decision hierarchy. This representation can be interpreted in terms of the valence-arousal model of emotion. We find that this model does not give a particularly good fit to the data: although the *arousal* dimension can be identified easily, *valence* is not well represented in the transformed data. We describe a new hierarchical classification technique whose structure is based on these calculated NMDS plots, which we call Data-Driven Dimensional Emotion Classification (3DEC) @. This new method is compared with the best of the four classifiers and a state-of-the-art classification method on all selected databases.

6.1 Motivation

Most of the contributors in the Interspeech 2009 Emotion Challenge (Schuller *et al.*, 2009b) employed binary classifiers arranged in a hierarchical structure for multiclass classification of the Aibo database. For example, Lee *et al.* (2009) and subsequently Lee *et al.* (2011) used a binary decision tree; at each node of the tree, either a binary Bayes logistic regression or support vector machine (SVM) classifier was trained (results are given for both of these). Based upon prior empirical testing, the authors placed negative *angry*/*emphatic* versus positive emotions at the root node (see their Figure 1), because these two groups of classes were highly discriminable, and they left the decision between *neutral* and *rest* until the end because of their high level of similarity. Planet *et al.*

(2009) used multiple classifiers with different hierarchical structures. In one of these, they used a binary classifier to distinguish between *neutral* and other classes at the first level and then used a multiclass classifier at the next level to separate the remainder of the classes. In another scheme, a cascade of binary classifiers was used, arranged in a waterfall structure in which at each level they separate the most populated class from the remainder. So for M classes, they had to train $M-1$ classifiers. Luengo *et al.* (2009) used a one-versus-rest scheme for training two different types of classifiers for each emotion and fused the results using a scoring scheme. Shaukat and Chen (2008) used the valence-arousal model to determine a hierarchical structure for classification of the Serbian emotional speech database. They acknowledge the fact that emotions in the *arousal* domain are easier to classify than the ones in *valence* domain by dividing the emotions into active and non-active emotions using a binary SVM classifier at the root node. They further divided the emotions depending upon the *valence* to give the final classification results.

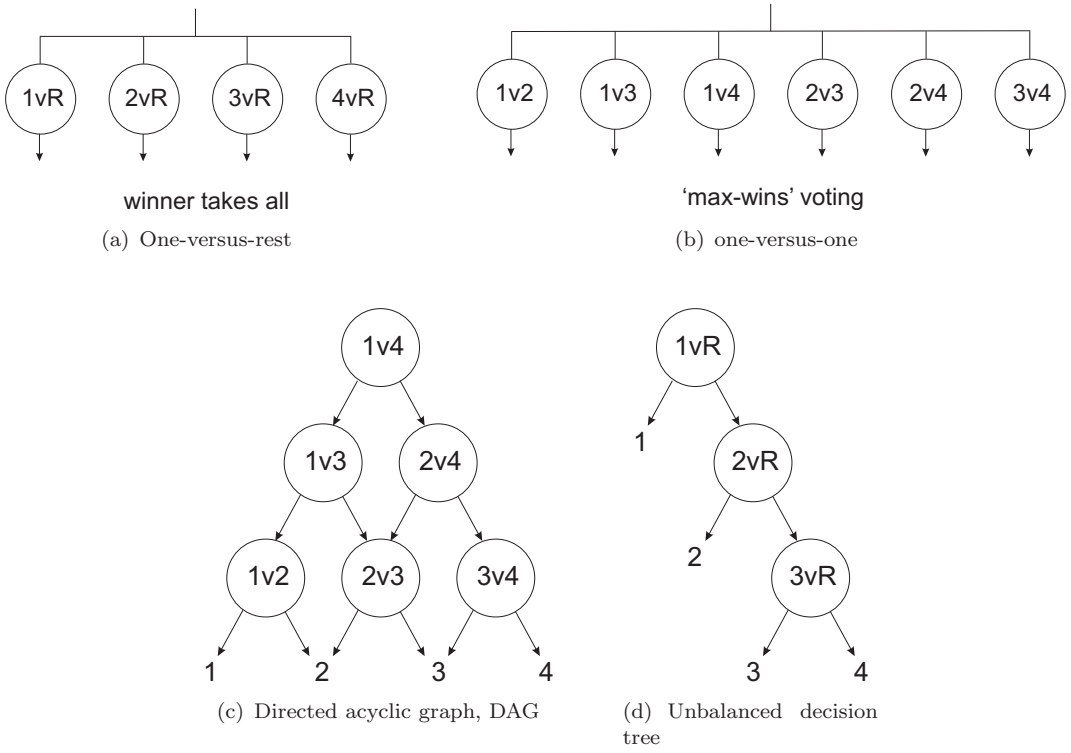
6.2 Four Methods of Multiclass Classification

Support vector machines were originally developed for binary classification; much subsequent work has been done to extend them to multiclass classification. There are two main approaches for multiclass classification algorithms: one considers all classes in one single optimisation function and the other breaks the problem down into several binary classifications. In this thesis, we have concentrated on the second approach exclusively. As an example, Hsu and Lin (2001) describe the one-versus-one (1v1) classifier, one-versus-rest (1vR) is described by Cristianini and Shawe-Taylor (2000), the directed acyclic graph (DAG) by Platt *et al.* (2000) and unbalanced decision trees (UDTs) by Ramanan *et al.* (2007). All of these approaches solve the multiclass classification problem by dividing it into a combination of binary problems. Two of these form hierarchical structures while the other two are non-hierarchical as shown in Figure 6.1.

In the next few sections, we give the details of the four methods that we have used for multiclass classification.

6.2.1 One-versus-Rest

The 1vR scheme applies a 'winner takes all' strategy. To classify M classes, this method constructs M binary classifiers in parallel, as shown in Figure 6.1(a) for $M = 4$. This combination can be seen as M classifiers in parallel. The jth classifier is trained with all the training data in the jth class given positive labels, and the rest given negative labels. Thus, given n training samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}^D$, $i = 1, 2, \ldots, n$ and $y_i \in \{1, 2, \ldots, M\}$, we have M binary classifiers each with



(a) One-versus-rest

(b) one-versus-one

(c) Directed acyclic graph, DAG

(d) Unbalanced decision tree

Figure 6.1: Various architectures for combination of binary classifiers for four classes: (a) One-versus-rest; (b) One-versus-one; (c) Directed acyclic graph; (d) Unbalanced decision tree.

a classification function J_j. An input test sample x_i is assigned to the class with the highest classification function value:

$$\text{class of } x_i = \underset{j=1,\ldots,M}{\operatorname{argmax}} J_j(x_i) \tag{6.1}$$

The benefit of the 1vR scheme is that we only have to train M classifiers. However, we have to deal with highly unbalanced training data across the binary classifiers, which compromises performance. To deal with this, down sampling is applied to the non-target class to make the learning task easier for the corresponding classifier.

6.2.2 One-versus-One

The 1v1 classifier uses a 'max-wins' voting strategy as illustrated in Figure 6.1(b) for the case of four classes. It constructs $M(M-1)/2$ binary classifiers in parallel, one for every possible pair of distinct classes. Each binary classifier J_{ij} is trained on the data from the ith and jth classes only. For a given test sample x_i, if classifier J_{ij} predicts that it belongs to class i, then the vote for class i is increased by one; otherwise the vote for class j is increased by one. At the end of this process, the voting strategy assigns the test sample to the highest scoring class. In this method, we do not have to handle highly

unbalanced classes as in the case of 1vR. However, we have to train more classifiers than for 1vR.

6.2.3 Directed Acyclic Graph (DAG)

The DAG method proposed by Platt *et al.* (2000) also has to train $M(M-1)/2$ binary classifiers for M classes. The training phase is the same as for 1v1; however, in the testing phase, it uses a rooted binary directed acyclic graph with $M(M-1)/2$ internal nodes and M leaves. Each node is a binary SVM classifier, J_{ij}, for ith and jth classes. For every test sample, starting at the root node, the sequence of binary decisions at each node determines a path to a leaf node that indicates the predicted class. Figure 6.1(c) shows the DAG architecture for the classification of four classes. In this scheme, usually there are several paths that can be taken to correctly classify a test sample. This adds redundancy in the structure which helps making the correct decisions. Each binary classifier has to deal with only two classes at a time which makes training easier and more effective as compared to 1vR.

6.2.4 Unbalanced Decision Tree (UDT)

The unbalanced decision tree (UDT) converts the 1vR scheme into a right-branching tree structure (Ramanan *et al.*, 2007). In this method, we have to train $(M-1)$ binary classifiers as compared to the M classifiers of 1vR. These are arranged in a cascade structure similar to that used by Planet *et al.* (2009). Starting at the root node, one selected class is evaluated against the remainder. The decision about which classifier to place at the root is made by taking the most separable class from the rest using the results of the 1vR scheme. Then, the UDT method proceeds to the next level by eliminating the class from the previous level from the training samples. That is, UDT uses a 'knock-out' strategy that, in the worst-case scenario, requires $(M-1)$ classifiers, and in the best case needs only one classifier. Figure 6.1(d) shows the architecture of UDT for classification of four classes. Class 1 has been chosen as the root node on the assumption that this class is maximally separable from the rest.

6.3 Classification Results using the Four Methods

Here we present the results of applying the above mentioned four multiclass classification schemes using features selected by Bord-Hard. Table 6.1(a) shows the UA and WA percentage accuracy obtained by SD–CV using the above-mentioned four multiclass classification methods on the four databases described in Section 3.4. In all cases reported in this chapter, linear SVMs with $C = 0.1$ have been used. The results of

Table 6.1: Unweighted and weighted average percentage accuracies for speaker dependent 10-fold and speaker independent leave one speaker out cross validation with four classifier schemes. Figures in brackets are standard deviations across the corresponding folds or speakers.

(a) Speaker-Dependent 10-fold Cross Validation

Database	**# classes**		**1vR**	**UDT**	**1v1**	**DAG**
DES	5	UA	59.6 (2.9)	69.6 (8.4)	69.6 (6.2)	**70.0** (8.8)
		WA	59.6 (2.9)	69.6 (8.4)	69.6 (6.2)	**70.0** (8.8)
Berlin	7	UA	84.7 (3.9)	88.4 (6.1)	88.2 (2.6)	**90.2** (5.3)
		WA	85.4 (3.8)	88.6 (6.1)	89.2 (2.4)	**90.6** (4.7)
Serbian	5	UA	92.3 (2.0)	94.0 (1.0)	94.2 (0.8)	**94.7** (1.0)
		WA	92.3 (2.0)	94.0 (1.0)	94.2 (0.8)	**94.7** (1.0)
Aibo-Mont	5	UA	43.7 (3.0)	43.2 (3.5)	**49.9** (2.4)	44.1 (2.4)
		WA	41.2 (3.0)	43.0 (3.2)	48.9 (2.7)	**49.5** (1.8)
Aibo-Ohm	5	UA	43.4 (3.4)	45.1 (2.1)	**53.3** (1.7)	42.1 (0.3)
		WA	43.0 (3.2)	57.9 (1.7)	52.8 (1.8)	**61.9** (0.7)

(b) Speaker-Independent Leave One Speaker Out

Database	**# classes**		**1vR**	**UDT**	**1v1**	**DAG**
DES	5	UA	42.7 (7.7)	47.3 (12.9)	49.6 (6.9)	**51.9** (9.2)
		WA	42.7 (7.7)	47.3 (12.9)	49.6 (6.9)	**51.9** (9.2)
Berlin	7	UA	74.5 (9.9)	75.5 (7.3)	77.4 (9.6)	**77.9** (7.8)
		WA	77.5 (8.1)	78.5 (5.4)	**79.9** (8.5)	79.5 (8.9)
Serbian	5	UA	77.9 (4.9)	72.6 (4.7)	**79.0** (5.9)	78.9 (5.4)
		WA	77.9 (4.9)	72.6 (4.7)	**79.0** (5.9)	78.9 (5.4)
Aibo-Mont	5	UA	38.8 (6.7)	38.7 (6.2)	39.4 (7.9)	**41.4** (6.1)
		WA	59.7 (6.8)	54.5 (4.7)	43.2 (10.5)	**65.8** (6.9)
Aibo-Ohm	5	UA	35.3 (6.9)	40.3 (2.1)	42.4 (8.5)	**44.0** (4.9)
		WA	61.3 (9.4)	54.6 (5.6)	43.5 (9.3)	**61.8** (8.8)

1v1 have already been reported in Chapter 5. Out of the four databases tested, the test samples in DES and Serbian databases are balanced for each emotional class. Therefore, UA and WA accuracies for these two are exactly the same. The Berlin database is not badly unbalanced and so there is only a small difference between UA and WA accuracies. On the other hand, Aibo is highly imbalanced for which we believe UA gives better representation of the performance of the classifier. Generally, of the four methods, 1v1 and DAG appear to work best with UDT not far behind and 1vR the poorest method.

To get more realistic results, we apply SI–CV on the selected databases. Table 6.1(b) shows UA and WA percentage accuracies for this speaker-independent case. Comparing these with Table 6.1(a), it is very clear that SI–CV classification results are much lower as compared to SD–CV, which is only to be expected. Again, 1v1 and DAG work best with 1vR the poorest of the methods.

Further insight into the performance of the classifiers can be gained by looking at the

Table 6.2: Confusion matrix for 5-class SI–CV emotion classification for the DES database using DAG classification.

Input		Neutral	Angry	Happy	Sad	Surprised
	Neutral	**16**	5	5	10	8
	Angry	3	**24**	6	0	11
	Happy	2	3	**22**	2	15
	Sad	7	0	1	**31**	5
	Surprised	3	6	11	4	**20**

confusion matrices. Table 6.2 shows the confusion matrix of SI–CV when classifying five emotions from the DES database using a DAG classifier achieving 51.9% average speaker-independent accuracy with 9.2 standard deviation over the four speakers in this database. This combination of database and classifier was chosen for illustration as DES is the hardest of the databases that is not badly imbalanced, and DAG is one of the two best performing classifiers. On the main diagonal are the number of correctly classified test samples for each test class whereas the off-diagonal numbers are the corresponding misclassified test samples. It is clear that *happy* and *surprised* are often confused. Similarly, *angry* is also confused with *surprised* emotion. All of these confused emotions have high arousal. We observed similar patterns in Hassan and Damper (2010, 2009) in which high-arousal emotions were always difficult to separate from each other using acoustic features only.

6.4 Visualising with Multidimensional Scaling

Understanding and interpreting confusion matrices can be difficult if we have a large number of classes. A visual representation of confusion can be much more helpful in interpreting the results. In this section, we look at some of the methods that can be used to visualise confusion matrices which make their interpretation much easier.

6.4.1 Heat Plots

Figure 6.2 shows the heat plot of the confusion matrix shown in Table 6.2. The blocks on the diagonal are high shades of red indicating the correctly classified samples. The stronger the shade of red, the easier it is to classify that class, e.g., *sad* is the strongest shade of red as compared to the other four classes which means that this is the easiest class to separate. In any row, if two blocks have the same colour, these are highly confused with each other.

Looking at the Figure 6.2, it is very clear that the two classes most confused with each other are *happy* and *surprised*. This is slightly different from looking at the confusion

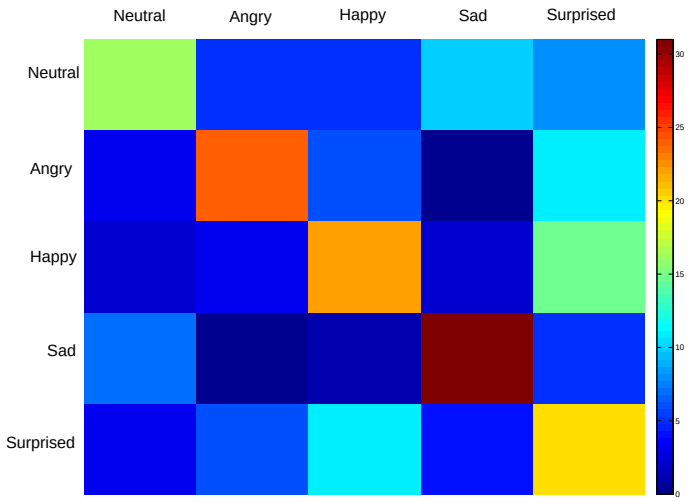


Figure 6.2: Heat plot of confusion matrix for 5-class SI–CVemotion classification for the DES database using DAG classification shown in Table 6.2.

matrix in which there were two pairs to be labelled as the most confused classes. The heat plots makes this decision much easier.

Although heat plots give a visual representation of a confusion matrix, there is another way of visualising the confusion matrix. It is by converting the similarities between different classes into distances between them. For this purpose, non-metric multi-dimensional scaling (NMDS) is very useful.

6.4.2 Non-Metric Multi-Dimensional Scaling

As an alternative to the analysis of the confusion matrix, we can use non-metric multi-dimensional scaling (NMDS) to visualise the patterns of similarities or dissimilarities among the input objects (speech tokens drawn from particular emotion classes in this case) in a low-dimensional space (Kruskal, 1964). The output of NMDS is a low dimensional visual plot; in our work, we have used two dimensions.

We use non-metric scaling as we do not consider the distances to be based on an interval or ratio scale, but view it as an ordinal scale in which monotonicity is maintained. This means that we will not assign exact meaning to the distances on the NMDS plot. Rather we will interpret the general disposition of the classes in the two-dimensional plot. For an ideal case where all the objects are equally distant, they should be arranged on a circle in two dimensions. Generally, the dimensions have a complex relation to the properties of the input objects, and so have to be interpreted subjectively and with care. Orientations are arbitrary.

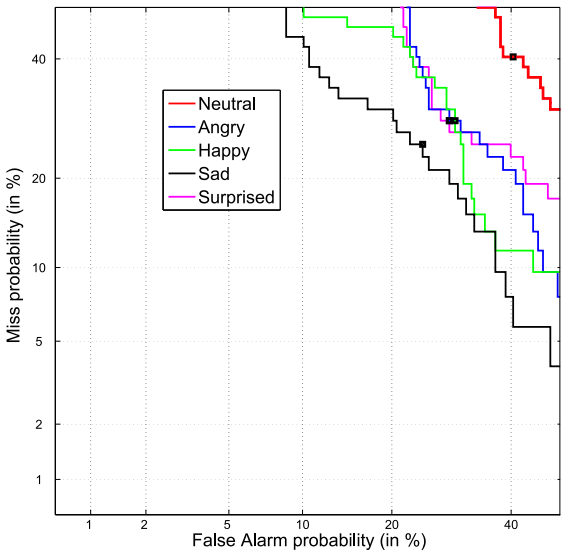
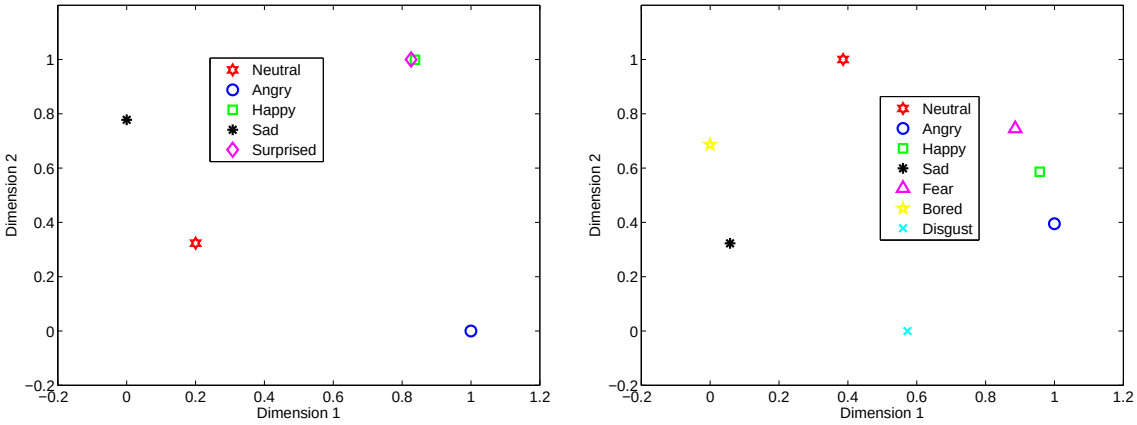


Figure 6.3: DET plot of confusion matrix for 5-class SI–CVemotion classification for the DES database using DAG classification shown in Table 6.2.

There are different ways of measuring similarities/dissimilarities between classes. Equal error rate (EER) from detection error tradeoff (DET) curves (Martin *et al.*, 1997) is often used for this purpose. Figure 6.3 shows the DET curves for the DES database. The dark squares in the figure are the points where the two types of errors are equal, i.e., equal error rate. From the plot it is clear that *sad* is the easiest class to separate and we get best accuracy for this class while for *neutral* class the results are the worst. The other three classes have similar accuracies but we can not tell from the DET plots which classes are going to be confused with each other.

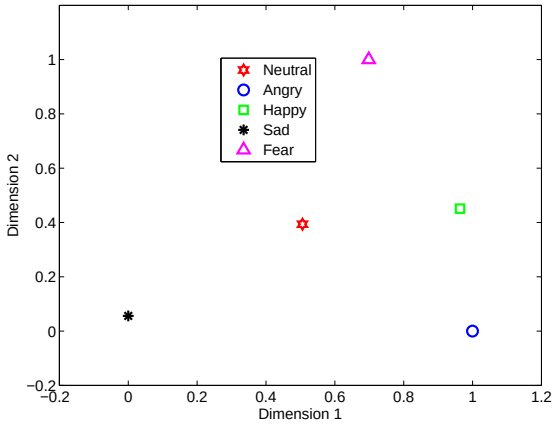
Since we have already obtained confusion data for the four methods in Section 6.3, these can be used directly. As we are using a non-metric method, the orientation of the plots using EER or using confusion matrices is identical. We have used the Statistical Package for Social Science (SPSS) for this transformation. Figure 6.4(a) shows the NMDS plot of the confusion matrix in Table 6.2 for DES database using the DAG classifier. The figure clearly illustrates that the two emotions *happy* and *surprised* are highly confused by placing them close together whereas the rest of the classes are well apart, consistent with our interpretation of the confusion matrix as above.

It is reasonable to view the construction of such NMDS plots as, in some sense, an empirical test of the valence-arousal and/or Russell's related circumplex model. From Figure 6.4(a), we can identify the arousal dimension in the north-west to south-east direction. That is, the high-arousal emotion *angry* and low-arousal emotion *sad* appear on the two extremes, while *neutral* is centrally placed. However, the valence dimension, which theoretically should be in the north-east to south-west direction, is nowhere near so clear since we observe positive valence *happy* oppositely placed to *neutral*. Similarly, it is not obvious that *surprised* should be classified as having strongly positive valence since

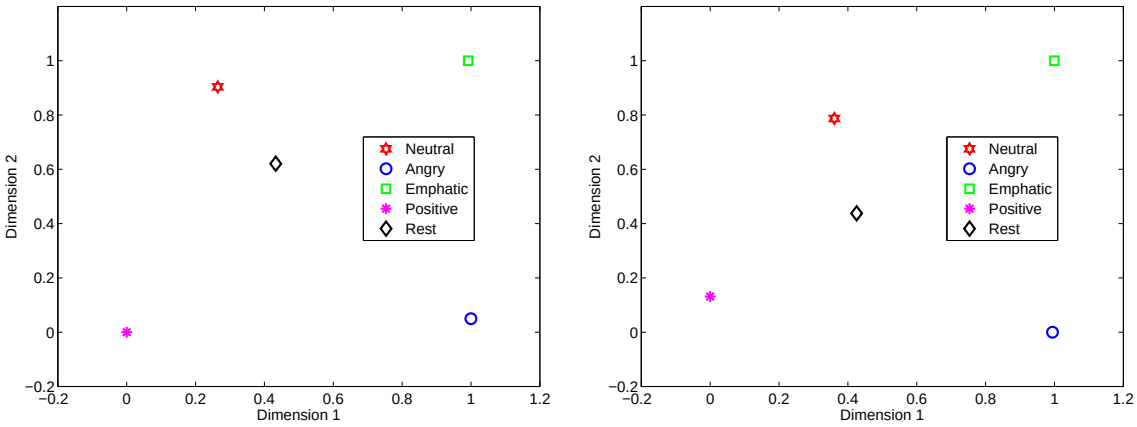


(a) DES

(b) Berlin



(c) Serbian



(d) Aibo-Mont

(e) Aibo-Ohm

Figure 6.4: NMDS representation of speaker-independent confusion matrices for (a) five-class DES database; (b) seven-class Berlin database; (c) five-class Serbian database; (d) five-class Aibo-Mont; (e) five-class Aibo-Ohm. Axis dimensions have arbitrary (non-metric) units.

this emotion could be either positive or negative. Figure 6.4(b) shows the NMDS plot for the confusion matrix of the Berlin data classified using DAG. Here, we can identify the arousal dimension in the east-west direction. That is, the high-arousal emotions *angry* and *happy* appear well to the east, medium-arousal *fear* is a little more centrally placed, and the low-arousal emotions *sad* and *bored* are well to the west. However, the valence dimension, which theoretically should be in the north-south direction, is nowhere near so clear since we observe negative valence *disgust* and *neutral* at opposite north-south extremes with positive valence *happy* in the middle. Similarly, positive valence *happy* is clustered with *angry*, which has negative valence. A similar pattern is seen in Figures 6.4(c) for the Serbian data, and for the Aibo-Mont and Aibo-Ohm data in Figures 6.4(d) and 6.4(e), respectively. (Note too the very good agreement between Mont and Ohm.) Overall, we observe the similar presence of an easily identifiable *arousal* dimension but there is no obvious *valence* dimension present in these plots.

Shaukat and Chen (2008) made a similar observation (i.e., that *arousal* is easier to distinguish from the acoustic data than *valence*) and this is why they separated arousal on the first stage of their hierarchical classifier. Schuller *et al.* (2010) also support this finding. So there seems to be general agreement on this point. An exception is the work of Batliner *et al.* (2008). Using a data-driven approach similar to that employed here, they found that something akin to *valence* seemed to be well represented in their speech data. But this was not the case for the *arousal* dimension, which had to be replaced by something they called 'interaction'—having to do with whether an emotion was 'addressing oneself' (e.g., *angry*) or the communication partner (e.g., reprimanding). The reason for this discrepancy is unknown; suffice to say that we are in accord with the general finding, which seems to point to a deficiency in the valence-arousal model. Although *valence* should be somehow represented in the data, we infer from our work that it is not in an orthogonal direction to *arousal* as is generally assumed. However, the original valence-arousal model was developed using perception models while our results have been obtained in the feature space. Therefore, these results are to be considered with care.

6.5 Data Driven Dimensional Emotion Classifier, 3DEC

As NMDS plots cluster the highly-confused classes together and place the easily separable classes away from each other, it should be possible to determine an appropriate structure for a hierarchical multiclass classification scheme based upon these plots. The benefit of such a scheme is that it is not motivated by theoretical adherence to the valence-arousal model, which according to the immediately preceding discussion seems to have its own deficiencies. Rather it is data driven; input data dictate how to organise the hierarchy of the classification scheme. The main idea is to structure the hierarchy so as to separate the highly-confused classes from the remainder at the first stage, and

progressively to apply the same strategy at later stages until all classes are separated. We call this scheme Data-Driven Dimensional Emotion Classification (3DEC).

6.5.1 Determining the 3DEC Structure

The 3DEC structure is determined in a data-driven fashion from NMDS plots, i.e., from confusion data. This immediately raises a practical issue, since proper estimation of the generalisation capabilities of the 3DEC scheme requires that the data used to determine the classifier confusions, and hence the hierarchical structure, need to be entirely separate from the data used subsequently for training and testing the classifier. (And, of course, the data for training and testing the classifier need to be similarly disjoint, as detailed in Section 6.3.) An excellent way to achieve this would be to use human confusion data to derive the NMDS plots. However, human confusion data are only available for the DES and Serbian databases. Hence, we decided to use a cross-validation approach to separate the training/testing data from the data used to determine confusions, as detailed in the following subsubsection. Confusions were computed using 1v1 as the base classifier as this performed well in the work reported in Section 6.3.

6.5.1.1 Data Partitioning for Determining Confusions

In the case of speaker-dependent recognition, the complete database is first partitioned into 10 folds. Each of the 10 folds is held-out in turn for testing, and the remaining 9 folds are used for determination of NMDS confusion plots. To keep the data disjoint and avoid getting over optimistic results, we have applied 9-fold CV to get the confusion matrices from the training folds which are used to derive the hierarchical structure of 3DEC classifier. This 3DEC classifier is then tested on the held out fold and the process is repeated for all the remaining k-folds.

For illustration, Figure 6.5 shows the NMDS plots found in this way for each of the 10 folds of the Aibo-Ohm database. There is clearly a high degree of similarity between the 10 plots, and this was also the case for the other databases, which is reassuring. Another relevant observation was that the NMDS plots found by the data-driven method were very similar to those found from human confusion data, where the latter were available. The plots for the remaining databases are given in Appendix B.

A similar approach was adopted for the speaker-independent case, except that the complete database was partitioned into s folds (i.e., as many folds as speakers) with each of the s folds being held-out in turn and the remaining $(s-1)$ folds being used for determination of NMDS plots. Again, we require to separate data for training the binary SVM classifiers from data used for deriving the confusions so we used $(s-1)$-fold CV.

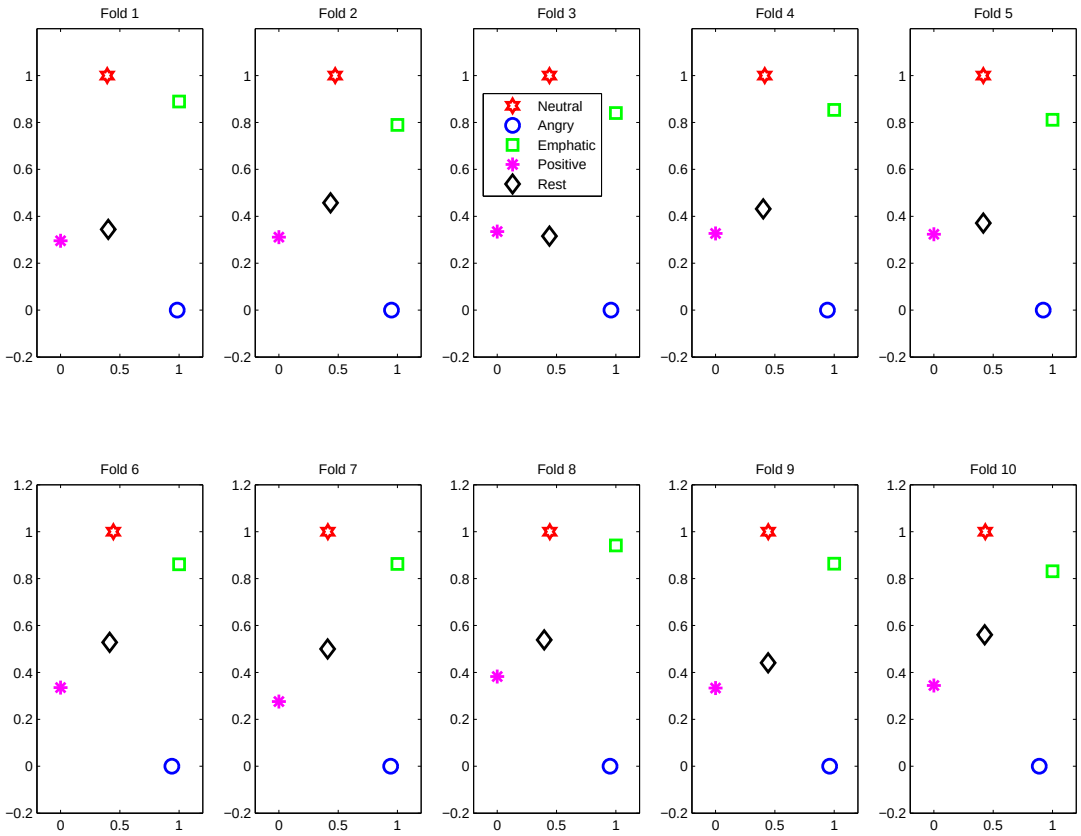


Figure 6.5: NMDS plots for the 10 folds of the Aibo-Ohm database in the speaker-dependent case.

6.5.1.2 Example 3DEC Structure

Considering Figure 6.5, it is clear that *neutral* and *emphatic* cluster together in the Aibo-Ohm data and are well separated from the remaining three classes. Therefore, the binary classifier at the root of the hierarchy for classifying this dataset is designated N&E vs A&P&R in Figure 6.6. Similarly, at the next stage, we separate *angry* from the remaining two classes, *positive* and *rest*, and we can also now separate *emphatic* from *neutral*. Finally, we separate *positive* and *rest*. Figure 6.6 shows the complete 3DEC hierarchical scheme for the Aibo-Ohm database. Specific data-driven classifiers for the other databases were built in the same way.

6.5.2 Performance Evaluation of 3DEC

We have evaluated the 3DEC method on all of the four databases. For comparison, we have chosen the best-performing of the four earlier multiclass classification methods described in Section 6.2 (which we judge to be DAG) and the structure based on valence-arousal theory described by Shaukat and Chen (2008)—hereafter denoted S+C. The binary classifiers are as described in Section 6.2. Both speaker-dependent and speaker-independent performance are considered.

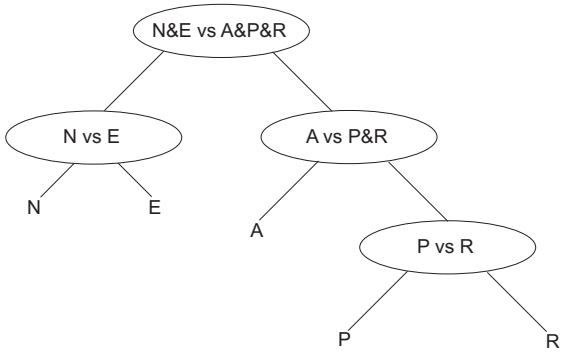


Figure 6.6: 3DEC scheme for five-class Aibo-Ohm database. Key: N – *neutral*; A – *angry*; E – *emphatic*; P – *positive*; R – *rest*.

Table 6.3: Percentage UA accuracies for SD–CV and SI–CV using linear SVMs for 3DEC model, DAG and the method of Shaukat and Chen (S+C). Bold font indicates the best accuracy for each classifier, database and speaker-dependent/independent test condition.

	SD–CV			**SI–CV**		
Database	**3DEC**	**DAG**	**S+C**	**3DEC**	**DAG**	**S+C**
DES	**74.6** (3.2)	70.0 (8.8)	70.8 (3.2)	**53.9** (8.2)	51.9 (9.2)	48.2 (10.0)
Berlin	**92.1** (3.1)	90.2 (5.3)	84.8 (3.8)	**79.5** (6.0)	77.9 (7.8)	76.3 (6.6)
Serbian	94.6 (0.8)	**94.7** (1.0)	94.3 (0.5)	**80.1** (4.1)	78.9 (5.4)	73.9 (10.0)
Aibo-Mont	**49.4** (3.3)	44.1 (2.4)	48.1 (2.3)	**43.4** (6.0)	41.4 (6.1)	41.6 (6.6)
Aibo-Ohm	**48.1** (1.4)	42.1 (1.3)	44.8 (2.1)	**44.8** (2.1)	44.0 (4.9)	43.6 (4.0)

We have already discussed that for unbalanced data, percentage WA accuracy gives unrealistic values. Therefore, table 6.3 only shows the percentage UA accuracy in each case. For each row of the table (i.e., each database), and for each speaker-dependent/independent condition, the best-performing method is indicated by highlighting its accuracy in bold font. Speaker-dependent results are inevitably higher than corresponding speaker-independent ones since the classifier can only profit from having access to some speaker-specific data. For the speaker-dependent case, there is no obvious winner. However, for the much more interesting and realistic speaker-independent case, 3DEC is the top-ranking method in all cases.

6.5.3 Statistical Analysis

To determine if there are significant differences in the performance of the three algorithms as tabulated in Table 6.3, we have used the Friedman two-way analysis of variance by ranks (Siegel and Castellan, 1988). In this case, there are $N = 5$ blocks, i.e., databases of emotional speech, and $k = 3$ treatments, i.e., classification algorithms. The null hypothesis H_0 is that the treatments have identical effects (i.e., all three algorithms perform equally well); the alternative hypothesis H_a is that at least one algorithm is

different from at least one other algorithm in its performance. The relevant test statistic is denoted χ_r^2 (since for N and k not too small, it is distributed as chi-square with $k-1$ degrees of freedom). Note that this is a non-parametric test (i.e., it does not make an assumption of normal distributions).

Applying the Friedman test to the SD–CV results gives $\chi_r^2 = 4.8$ corresponding to $p > 0.05$ but $p < 0.20$. Hence, we do not have a very strong basis to reject H_0 in favour of H_a, and conclude that there is no statistically significant difference at 5% level in speaker-dependent performance between 3DEC, DAG and S+C. However, for the much more interesting speaker-independent case, we obtain $\chi_r^2 = 8.4$ with $p < 0.02$ leading us to accept H_a, and conclude that there are indeed statistically significant differences in performance between the three methods at better than the 2% level. This being so, it is now legitimate to apply a paired t-test to the individual treatments (under the assumption of normal distributions). This shows that 3DEC significantly outperforms DAG ($p = 0.0009$, one-tail test), which in turn significantly outperforms S+C ($p = 0.0115$, one-tail test).

6.6 Comparison with State-of-the-Art

In general, the results in Table 6.3 are not directly comparable with state-of-the-art results reported elsewhere in the literature for the same databases. This is because, although the databases are the same, the selected feature sets, classifiers and training/test regimes (e.g., the specific folds used in cross-validation) mostly differ, sometimes very markedly. Nonetheless, we think it is valuable and instructive to make some qualitative comparisons between the current results and the state-of-the-art.

In Table 6.4, we have summarised —to the best of our knowledge—the state-of-the-art performance on the DES, Berlin, Serbian and Aibo databases together with the classifiers and the methods used to achieve them. The SD–CV results of Shami and Verhelst (2007) of 64.9% and 80.7% WA on DES and Berlin, respectively, compare with 74.6% and 92.1% UA for 3DEC in this work. It seems unlikely that a difference between 64.9% and 74.6% could be anything other than a strong pointer to the superiority of 3DEC over Shami and Verhelst's method. We are not aware of any other authors having published SI–CV results for DES. The best speaker-independent performance on Berlin of 85.6% obtained by Schuller *et al.* (2009a) compares to our figure using 3DEC of 79.5%. (Schuller *et al.*'s corresponding UA value was 84.6%.) However, there were some procedural differences (e.g., they used cepstral mean normalisation, a polynomial kernel for SVM classifier and balancing of the training partition).

In the case of SI–CV , Shaukat and Chen (2008) have obtained 89.0% on the Serbian database, some way above our value of 80.1%. There could, however, be many reasons for this, related to differences in evaluation regime like normalisation and discretisation

Table 6.4: State-of-the-art results on the four databases studied here. Accuracy is reported on the total number of emotion classes (which appears in brackets in the first column). All accuracies are weighted by the number of test tokens in each class.

	Source	**Accuracy (%)**	**Method**
DES (5)	Shami and Verhelst (2007)	64.9 WA	SD–CV with SVM and RBF kernel
Berlin (7)	Shami and Verhelst (2007)	80.7 WA	SD–CV with SVM and RBF kernel
	Schuller *et al.* (2009a)	84.6 UA, 85.6 WA	SI–CVwith SVM and polynomial kernel
Serbian (5)	Shaukat and Chen (2008)	89.0 WA	SI–CV with SVM and polynomial kernel
Aibo (5)	Lee *et al.* (2011)	41.6 UA, 39.9 WA	For five classes, training on Ohm and Testing on Mont using hierarchical structure of binary decision tree of Bayes logistic regression and SVMs
	Schuller *et al.* (2009b)	67.7 UA, 65.5 WA	For two classes, training on Ohm and Testing on Mont with sample balancing and linear SVMs

of features, rather than substantive differences in the performance of the algorithms. Most tellingly, the *direct* comparison of the 3DEC method with Shaukat and Chen's method reported in the previous section shows that 3DEC performs significantly better overall.

Regarding the spontaneous Aibo data, it is hard to make meaningful comparisons between our speaker-independent results and those of Lee *et al.* (2011), since the train/test conditions are very different. Whereas we have used SD–CV and SI–CV on Aibo-Mont and Aibo-Ohm separately, Lee *et al.* trained on Ohm and tested on Aibo-Mont in producing their result of 41.6% UA accuracy. It seems reasonable to suppose that training and testing on data from within the same school (as we have done) might be easier than cross-school train-and-test, but beyond this it is difficult to speculate. We will test this scenario in the next chapter and do the relevant comparisons.

6.7 Summary

In this chapter, we have considered various ways of extending binary support vector machines to suit them for multiclass classification of emotions from acoustic features using four databases of emotional speech: three acted (Danish, German and Serbian languages) and one spontaneous (German). We have considered both speaker-dependent

and speaker-independent performance assessed by cross-validation, with results presented as the UA and WA performance measures. The four methods studied are the more or less standard one-versus-one and one-versus-rest approaches, plus two methods based on a hierarchical structure of classifiers, namely the directed acyclic graph (DAG) method of Platt *et al.* (2000) and the unbalanced decision tree (UDT) of Ramanan *et al.* (2007). Of the four methods, we find that DAG and 1v1 appear to work best, with UDT not far behind and 1vR the poorest.

Subsequently, we have used non-metric multidimensional scaling (NMDS) to visualise the acoustic data in two-dimensions, and attempted to interpret this visualisation in terms of the influential valence-arousal model of emotion. We find that the *arousal* dimension can be quite well identified in the acoustic data, but *valence* can not. Further, the form of the NMDS plots suggests a way to determine an appropriate structure for a hierarchical multiclass classifier, which we call 3DEC. This new classifier is compared with the DAG method and a state-of-the-art approach due to Shaukat and Chen (2008). For the speaker-dependent case, there is no strong evidence in favour of the 3DEC scheme. For the much more interesting and realistic speaker-independent case, however, 3DEC is the top-ranking method in all cases and the difference between it and the next best competitor is statistically significant ($p \sim 0.001$). In the next chapter, we extend these methods to inter–database (and therefore cross-language) classification and discuss the methods that can be used to compensate for the speaker and recording environment differences.

Chapter 7

Inter–Database Classification

In the last chapter, we have discussed several multiclass classification methods for emotion recognition. By applying standard SD–CV or SI–CV on a single database, many variables like microphone, room acoustics, noise and language remain constant. However, these results cannot be generalised onto other databases which will have different speakers and settings. These differences will have a big impact on the performance of the SER classifiers. Thus formal methods need to be applied to minimise the differences due to different speakers and acoustic environments between the training and testing data. Building these classifiers that generalise across varied acoustic conditions is highly desired for SER systems.

Within database (or intra–database) classification does not present such a strong challenge of varied environment and speakers that a commercial SER system will have to face. However, inter–database classification tests, i.e., training on one database while testing on another which has been recorded in entirely different acoustic environment, pose realistic and challenging testing conditions. Since most of the emotional speech databases do not provide clear separation of the training and testing datasets, another great advantage of inter–database classification experiments is the clear separation of the training and testing datasets. This makes the results reproducible and comparable. By performing inter–database emotion classification, we ask two questions: the first one is regarding the generalisation capabilities of SER system, as discussed, and the second one is '*Are emotions universal across different languages*?', especially when we have two databases with different languages.

Surprisingly, this area of research has been only very recently picked up by the machine learning community. Initial results show that it is indeed a very difficult task. In this chapter, we give an overview of current trends and approaches being used for this task. Then we apply well known methods used in automatic speech recognition (ASR) systems to improve the generalisation capabilities of SER and to the problem of intra– and inter–database emotion classification. These well known methods are maximum

likelihood linear regression (MLLR) using vocal tract length normalisation (VTLN) and cepstral mean normalisation (CMN).

In this chapter, we test a few methods from the emerging field of *transfer learning* to improve the generalisation of SER. Although, these algorithms have been developed in different scenarios, we identify the differences caused by varied speakers and recording acoustic environments for intra– and inter–database emotion classification as a special case of transfer learning. We believe that these differences can be modelled as a covariate shift. This covariate shift is then minimised using importance weights (IW) obtained from the training and testing data using importance weighting algorithms (IW-algorithms).

In this chapter, we give a thorough background of these algorithms and compare their performance on artificial data. We then compare their performance with standard CMN and MLLR algorithms for intra and inter–database emotion classification. Our comparative results show that IW-algorithms can be successfully used to compensate for the training and testing data difference caused by different speakers and acoustic environments. Our inter–database experiments also show that there are aspects of emotions that are *universal* across languages. This statement is true at least for the languages that belong to same geographical region.

7.1 Problem Statement and Motivation

Evidence for the universality of emotions comes from the experiments done by psychologists on cross-lingual emotion detection. Tickle (2000) asked Japanese and American English listeners to recognise five emotions expressed by native Japanese and American speakers without any semantic information. The experiments were done for *happy*, *sad*, *angry*, *fearful* and *calm* emotions and speakers from each native area were asked to recognise emotions from their own language and from the corresponding foreign language as well. The results obtained were above random accuracy indicating that emotions can be universal even across different languages. Abelin and Allwood (2000) have reported on similar experiments using Swedish and English languages. They also found that people were able to recognise emotions from speech although they did not know or understand the language. This does show that there are aspects of emotions that are universal and culture independent.

7.1.1 Problems with Inconsistent Databases

We have previously discussed some of the important problems with the current databases in Section 2.5.1. For inter–database classification all of these issues are even more pronounced. One of the biggest problems, other than environment and language mismatch,

will be the mismatch of annotation of class labels across databases. For inter–database classification, this posses a big problem, as the training and testing datasets must use the same class labels. This is specially complicated when some databases are labelled on the dimensional scale (Sensitive artificial listener database (Douglas-Cowie *et al.*, 2007)). To cater for these problems, one has to either find those databases which have few classes common among each other or map the emotions on some basic dimensions which are common among the two.

There is an argument for making efforts to create databases that consider these drawbacks. It is not very easy to compensate for all these factors by using one method. However, we can try to compensate for a few of them by using some systematic methods and try to improve the generalisation of the SER.

7.1.2 State-of-the-Art in Inter-Database Emotion Recognition

One of the first attempts to apply machine learning on inter-database emotion recognition was by Shami and Verhelst (2007). They used four databases in their work: namely DES, Berlin, BabyEars and Kismet. They did inter-database experiments on DES–Berlin and BabyEars–Kismet combination on four and three emotion classes common between these database sets respectively. They only got slightly above random accuracy using a SVM classifier. Although the results were not very encouraging, they proved that it is possible to perform inter-database emotion classification by using acoustic features.

More recently, Eyben *et al.* (2010) used four databases for 'positive', 'negative' and 'neutral' *valence* recognition using SmartKom, Aibo, Sensitive Artificial Listener (SAL) and Vera am Mittag (VAM) databases. All of these databases belong to the Germanic family of languages. They applied leave-one-database-out cross validation and on average achieved 53.4% unweighted average accuracy on all four databases using SVM classifiers. This accuracy is much higher than chance level (33.3%) on the notoriously difficult dimension of emotions (*valence*) which is very encouraging.

Schuller *et al.* (2010) did much extensive work and performed inter-database classification on six databases to recognise *valence* and *arousal* as well as the common emotion classes between the databases. They got much better results than random. However, they noted that for inter-database experiments, the accuracy decreases as the number of classes is increased. They have claimed that normalising the data for each speaker to a mean of zero and standard deviation of one compensates for some of the differences between the corresponding databases.

Lefter *et al.* (2010) used the combined data from four different databases to recognise common emotions. Out of the used databases, three contain acted speech while one has spontaneous speech data. Equal error rate on three common emotion classes (*anger*,

happiness and *sadness*) was reported. In their inter–database classification tests, they got best results for the *sadness* emotion class.

Similarly, Schuller *et al.* (2011b) used six current emotional speech databases for recognising *valence* and *arousal* by using leave-one-database-out cross validation method. They presented their results by combining the training data of all the databases to train a single classifier as well as combining the results of several classifiers, each trained on a different database, by majority voting. They found that majority voting increased the overall UA accuracy.

To the best of our knowledge, these are the only researchers who have done experiments on inter–database emotion classification. This small number of papers in this area of research shows that it has been usually overlooked by the emotion recognition community. We think that this problem surfaced after Interspeech 2009 emotion challenge in which two similar databases were used for training and testing the classifiers. An important observation is that none of the papers mentioned earlier explicitly compensates for the differences between the training and testing conditions for the inter–database emotion recognition, other than Schuller *et al.* (2010) who has applied speaker normalisation to compensate for this mismatch.

However, this problem of mismatch has long been recognised in the ASR research community and there are well known methods to compensate for these differences. In the next few sections, we will thoroughly review these methods used in ASR systems to reduce the effect of different speakers and acoustic environments.

7.2 Traditional Methods for Adaptation

Mel frequency cepstral coefficients (MFCCs) are usually the feature of choice for speech and speaker recognition tasks. Details of MFCC extraction have been given in Section 4.2.4. The reasons why MFCC features have been widely accepted and used in SER are their relatively good performance even under noisy conditions. Secondly, good robustness to environmental and speaker changes can be attained by adequate preprocessing. Usually, cepstral mean normalisation (CMN) is applied to compensate for the differences between the training and testing acoustic environments. It is achieved by subtracting the mean value of the cepstrum from the transformed signal. Vocal tract length normalisation (VTLN) is a method which is used to normalise the speaker differences caused by different vocal tract lengths. It is achieved by warping the frequency axis of the power spectrum by a speaker specific parameter. Similarly, maximum likelihood linear regression (MLLR) is used to adapt the previously calculated acoustic models to the particular speaker. All ASR systems apply some or a combination of these methods to adapt the acoustic features and ASR models towards the new environment

and unknown speakers. In the next section we give some description of these methods and how they are applied.

7.2.1 Cepstral Mean Normalisation

The idea of cepstral mean normalisation (CMN) stems from homomorphic analysis of speech, introduced in Oppenheim and Schafer (1968). Let us consider the speech signal $x(t)$ consisting of the noise-like source signal $n(t)$ which is convolved with the impulse response of the vocal tract $h_v(t)$. If we consider the environment of the speaker as another filter with an impulse response $h_e(t)$, then the speech signal can be written as:

$$x(t) = n(t) * h_v(t) * h_e(t) \tag{7.1}$$

The $h_e(t)$ is the unwanted part of the signal which can vary between training and testing setups and is going to affect the quality of extracted features. The idea of homomorphic analysis is that signals that are convolved in time domain will have a multiplicative effect in the frequency domain and additive effect in the cepstral domain because of the log function. The cepstrum of a signal is formally defined as '*power spectrum of the logarithm of the power spectrum*' (Noll, 1967) which can be mathematically written as:

$$x_{cep}(\tau) = \left|\mathcal{F}\{\log|\mathcal{F}\{x(t)\}|^2\}\right|^2 \tag{7.2}$$

where τ is the cepstrum equivalent of the time axis. The environment (h_e) is considered constant in comparison to the speech signal, therefore, it can be easily estimated by taking the mean of the transformed cepstral signal over several frames.

The impulse response of the environment can then be removed by subtracting the mean of the transformed signal.

$$\hat{x}_{cep}(\tau, k) = x_{cep}(\tau, k) - \bar{x}_{cep}(\tau) \tag{7.3}$$

where k denotes the frame index. To apply it to real time applications, an adaptive estimate of mean can be used.

7.2.2 Maximum Likelihood Linear Regression

Most recent ASR systems use hidden Markov models (HMMs) with Gaussian mixture emission probabilities for modelling speech of a particular speaker. A Gaussian mixture model is a weighted sum of N component Gaussian densities:

$$g(x) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(x|\boldsymbol{\mu}_i \boldsymbol{\Sigma}_i) \tag{7.4}$$

where λ_i are the weights and $\mathcal{N}(x|\boldsymbol{\mu}_i\boldsymbol{\Sigma}_i)$ are the component Gaussian densities with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Each component density is a Gaussian function of the form:

$$\mathcal{N}(x|\boldsymbol{\mu}_i\boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma_i}|}} \exp\left\{-\frac{1}{2}(x-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(x-\boldsymbol{\mu}_i)\right\} \tag{7.5}$$

To make the system adaptable to the new speaker, instead of learning a separate model for each speaker a general model is learnt from a large number of speakers from the available training data. Such models which generalise over all training speakers are called universal background models (UBM). When a new test speaker comes, maximum likelihood linear regression (MLLR) is used to calculate a linear transformation between UBM and the newly calculated GMMs. This linear transformation is calculated to maximise the likelihood of the adaptation data. For a given GMM model with mean vector $\boldsymbol{\mu}$, a new mean $\hat{\boldsymbol{\mu}}$ can be calculated such that:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}^{-1}\boldsymbol{\mu} + \boldsymbol{b} \tag{7.6}$$

where $\boldsymbol{A}$ is the transformation matrix and $\boldsymbol{b}$ is a bias vector. This linear transformation can be used to adapt the UBM for the new speaker or can be directly used to make decisions as done by Campbell *et al.* (2006). The same transformation matrix $\boldsymbol{A}$ can be used to transform $\boldsymbol{\Sigma}$. So all in all, we are linearly transforming the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of GMM's by matrix $\boldsymbol{A}$. This method was proposed by Leggetter and Woodland (1995) and is also known as constrained MLLR. Gales and Woodland (1996) introduced an extension to MLLR in which covariance matrices can also be adapted independently of the means. In our work, we have only considered the constrained MLLR.

7.2.3 Vocal Tract Length Normalisation

A major variability in the speech signal comes from the speaker specific vocal tract length. The vocal tract of females is, on average, 2-3 cm shorter than in men. The positions of spectral formant peaks of a given sound sample are inversely proportional to the length of the vocal tract. Due to this relation, the formant frequencies for females are 15-20% higher than male speakers. Any speaker and gender independent ASR system must be able to handle a range of speakers which includes both male and female speakers. Vocal tract length normalisation (VTLN) is a common technique used in ASR systems in order to minimise the inter-speaker variation. This technique scales the frequency axis of the acoustic features by introducing a speaker specific warping or normalising factor.

Linear, piecewise linear, bilinear and quadratic functions have been used for warping the frequency scale (Pitz and Ney, 2005). Figure 7.1 shows an example of a quadratic

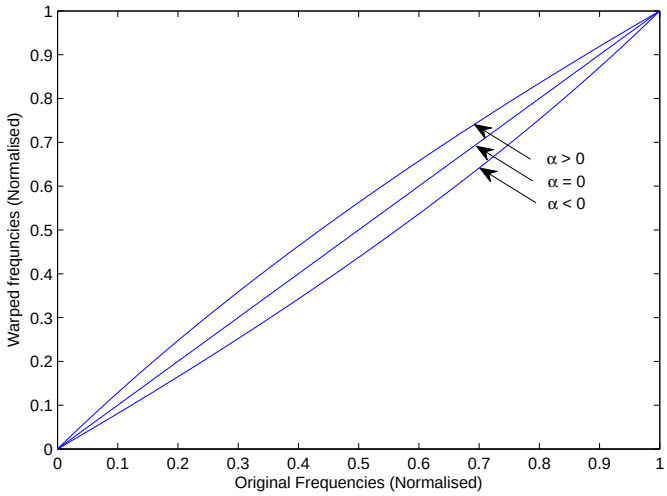


Figure 7.1: Example of quadratic VTLN warping function for $\alpha = -0.8, 0, +0.8$.

VTLN warping function with $\alpha = -0.8, 0, +0.8$. The warping factor α is calculated for each speaker independently.

There are two methods for calculating the warping factor α: model-based and feature-based. With model-based warping factor estimation, the idea is to build models with all possible warping factors. During the time of decoding/testing, select the model and warping factor, that gives maximum likelihood.

In the case of feature based methods, the fact that formants are inversely proportional to the length of vocal tract is used to estimate α. Eide and Gish (1996) proposed to use only the third formant for estimating α while Gouvêa and Stern (1997) suggested to use all first three formants for the estimation. Usually only the first two formants are enough to estimate the warping factor. The main idea is to find the slope of the line passing through the first and second formants of the UBM reference speaker and the new target speaker (Kabir *et al.*, 2010). This can be explained mathematically as follows:

$$\begin{aligned} F_{1_{ref}} &= \alpha_i F_{1_i} \\ F_{2_{ref}} &= \alpha_i F_{2_i} \end{aligned} \tag{7.7}$$

where F_{1_i} and F_{2_i} are the first and second formant of the ith speaker. The Euclidean distance between the formants of the reference speaker and the target speaker is given by:

$$d = \sqrt{(F_{1_{ref}} - \alpha_i F_{1_i})^2 + (F_{2_{ref}} - \alpha_i F_{2_i})^2} \tag{7.8}$$

$$d^2 = a\alpha_i^2 - 2b\alpha_i + c \tag{7.9}$$

where

$$\begin{cases} a &= (F_{1_i})^2 + (F_{2_i})^2 \\ b &= F_{1_i}F_{1_{ref}} + F_{2_i}F_{2_{ref}} \\ c &= (F_{1_{ref}})^2 + (F_{2_{ref}})^2 \end{cases} \tag{7.10}$$

Differentiating with respect to α_i and equating to zero gives:

$$\alpha_i = b/a \tag{7.11}$$

which is the normalising factor for ith speaker.

Pitz and Ney (2005) have shown that VTLN warping with a function (g_α) is simply a linear transformation of the cepstral coefficients of the unwarped spectrum with transformation matrix $\boldsymbol{A}(\alpha)$. This matrix only depends upon the warping factor α. They further showed the equivalence of VTLN and constrained MLLR by showing that linear transformation of the observation vector x by transformation matrix $\boldsymbol{A}$ is equivalent to a linear transformation of the mean vector $\boldsymbol{\mu}$ and appropriate transformation of the covariance matrix $\boldsymbol{\Sigma}$ for the GMMs. This is shown in the following equations:

$$x \rightarrow y = \boldsymbol{A}x \tag{7.12}$$

$$\mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \rightarrow \mathcal{N}(y|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{7.13}$$

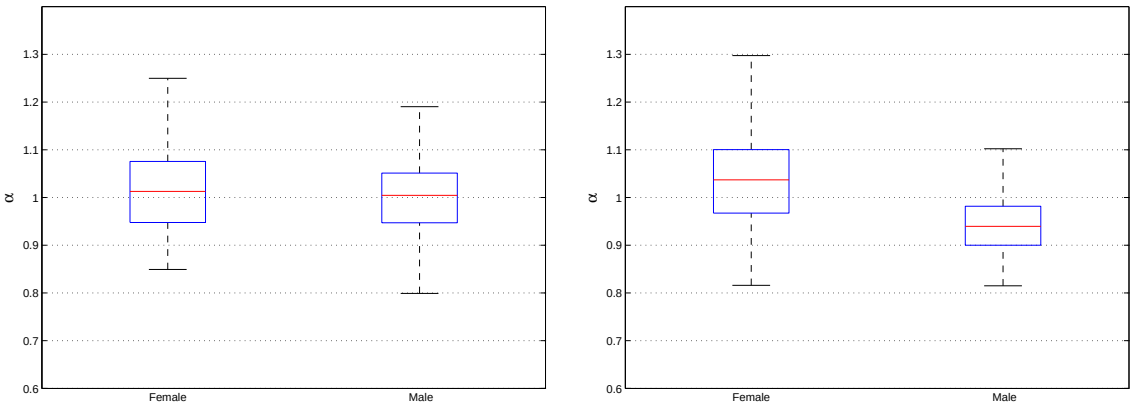
$$= \mathcal{N}(x|\boldsymbol{A}^{-1}\boldsymbol{\mu}, \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-1^T}) \tag{7.14}$$

$$= \mathcal{N}(x|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \tag{7.15}$$

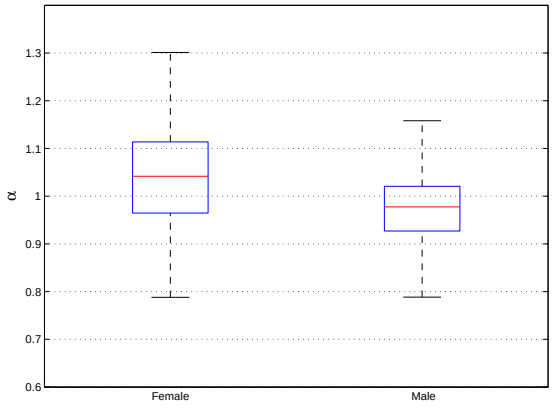
where $\hat{\boldsymbol{\mu}} = \boldsymbol{A}^{-1}\boldsymbol{\mu}$ and $\hat{\boldsymbol{\Sigma}} = \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-1^T}$. Therefore, VTLN is the constrained case of MLLR with only one free parameter, the warping factor α. Due to the ease of implementation of VTLN and its direct application to SVM classifiers, we have decided to use VTLN based constrained MLLR in our experiments.

Figure 7.2 shows the box plots for VTLN warping factor (α) calculated for all of the selected databases using Equation 7.11. The difference between the values of α for female and male speakers is quite clear for the databases which contain adult speakers. As the vocal tract is not fully developed in the children, we do not see much difference between male and female speakers for Aibo databases.

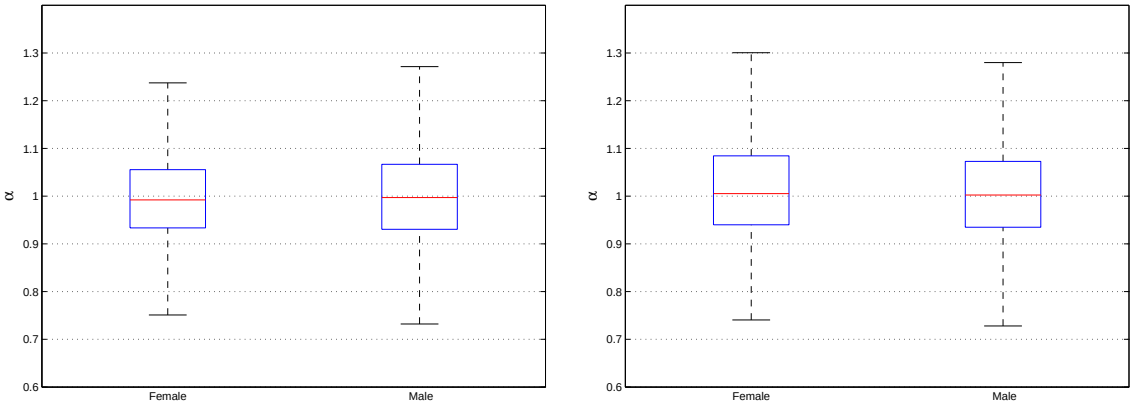
These two methods i.e., CMN and VTLN based MLLR, can be used to reduce the differences between the training and testing data induced by speaker and environment variations. They are standard methods already known to the speech recognition community. In the next few sections, we introduce transfer learning and some of its algorithms that we propose to be used to efficiently do the same task of reducing the mismatch between training and testing data samples.



(a) DES (b) Berlin



(c) Serbian



(d) Aibo-Mont (e) Aibo-Ohm

Figure 7.2: Box plots for VTLN warping factor α for the two genders for (a) five-class DES database; (b) seven-class Berlin database; (c) five-class Serbian database; (d) five-class Aibo-Mont and (e) Aibo-Ohm databases.

7.3 Transfer Learning

In 2005, DARPA gave a new mission for transfer learning: "the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks" (Pan and Yang, 2009). The main idea behind transfer learning is to borrow the knowledge gained from some related task or domain to help a machine learning algorithm to achieve better performance on the task of interest. It is motivated by human beings who can transfer knowledge from one domain to another. As an example, human beings may find learning to play squash is easier if one has already learned to play tennis. Similarly, learning to play checkers might facilitate learning to play chess. What is happening in human beings is that we are transferring the knowledge from one domain to another.

One of the implicit assumptions of any system developed using supervised machine learning techniques is that all the test and training samples are independent and identically distributed (i.i.d) and follow the same distribution in the training and testing data. However this fundamental assumption is quite often violated in practice. The conditions under which the system is developed are most likely not going to be the same as those under which they are deployed. This shift in the training and test data distribution is generally not considered during the building phase of the systems and is one of the main reasons why machine learning methods do not perform as well as expected in the real world. This problem was recognised by the speech recognition community and several methods have been developed to compensate for the channel distortion and environment mismatch between the training and testing conditions. However, it has only recently been recognised by the machine learning community under the general topic of *transfer learning*.

In a machine learning scenario, obtaining labelled data in the domain of interest is usually very expensive. It would be nice to be able to use the labelled training data from a related domain on the current domain of interest. The need to transfer knowledge may also arise when the data can out-date very quickly. As an example consider an intrusion detection system for which labelled data obtained at one time will not be valid after six months when the attack patterns have changed. Similarly, in brain computer interfacing, the data collected in one session will not be similar to another session when the participants' state of mind has changed. In such a case, knowledge transfer from one or more related domains to another target domain becomes very desirable. Formally transfer learning is defined as:

> Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T()$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

Table 7.1: Details of traditional machine learning and transfer learning settings.

Learning Setting		**Source and Target Domains**	**Source and Target Tasks**
Traditional Learning		$\mathcal{D}_S = \mathcal{D}_T$	$\mathcal{T}_S = \mathcal{T}_T$
Transfer Learning	Inductive	$\mathcal{D}_S = \mathcal{D}_T$	$\mathcal{T}_S \neq \mathcal{T}_T$
	Transductive	$\mathcal{D}_S \neq \mathcal{D}_T$	$\mathcal{T}_S = \mathcal{T}_T$
	Unsupervised	$\mathcal{D}_S \neq \mathcal{D}_T$	$\mathcal{T}_S \neq \mathcal{T}_T$

Transfer learning can be categorised into three distinct types: inductive, transductive and unsupervised transfer learning.

Inductive Transfer Learning In this setting, the target task is different from the source task but the domains of the two are the same.

Transductive Transfer Learning In this setting, the source and target domains are different but related; however, the tasks are the same.

Unsupervised Transfer Learning In this setting, both the domain and task are related but different from each other for source and target problems.

A brief summary of these three settings of transfer learning is given in Table 7.1 in comparison with traditional machine learning. For further details about these methods readers are referred to Pan and Yang (2009) in which all of these topics are dealt with in complete detail.

In this thesis, we are interested in transductive learning as the task of learning emotion classes is same for the training and testing data, while the data has a shift between them because of different speakers, environments and databases used for training and testing. In the next sections we identify these differences as covariate shift which is a special case of transductive learning and introduce some methods used in transfer learning to compensate for this shift.

7.3.1 Introduction to Covariate Shift

In standard supervised learning conditions, input observation (x) is independently drawn from the input distribution $p(x)$ and the corresponding label (y) from the conditional distribution $p(y|x = x_i)$, both in training and testing phase. A situation where the data is assumed to be generated by the model $p(y|x)p(x)$, where $p(x)$ changes between training and testing data is called covariate shift. There are many scenarios in real world applications where covariate shift is likely to occur. It mainly occurs because of the mismatch between the environments in which the training and the test data are collected. Some examples of such scenarios are spam filtering, online document classification, language processing, brain-computer interface and intrusion detection

systems. In all of these examples, data collected at one time may not be an accurate representation of the data collected at another time.

Let $\mathcal{X}$ be the input domain and suppose we are given i.i.d. training samples $\{x_i^{tr}\}_{i=1}^{n_{tr}}$ from the training distribution with density $p_{tr}(x)$ and i.i.d. test samples $\{x_i^{te}\}_{i=1}^{n_{te}}$ from a test distribution with density $p_{te}(x)$:

$$\{x_i^{tr}\}_{i=1}^{n_{tr}} \xleftarrow{i.i.d.} p_{tr}(x)$$
$$\{x_i^{te}\}_{i=1}^{n_{te}} \xleftarrow{i.i.d.} p_{te}(x)$$

We also assume that the input data densities are strictly positive. In the case of covariate shift, the input densities differ in training and testing, while the conditional distribution remains the same. In such a condition, we would like to assign more weight to those training samples that are most similar to those in the test set, and less weight to those that rarely occur in the test data. This method of weighting the input data based upon the test data is called importance weighting (IW). The goal is to estimate importance weights (β) from x_i^{tr} and x_i^{te} by taking the ratio of training and testing densities:

$$\beta(x) = \frac{p_{te}(x)}{p_{tr}(x)} \tag{7.16}$$

which is non-negative by definition. The main idea is to give more weight to those training data samples that give a better representation of test data. Similarly, less weight is given to those data samples that give least information about the test data. By doing so we push the learning algorithm towards the more important regions in input space. It is clear that the regions where the test distribution is denser would yield a large value of β, thus shifting the learning algorithm towards those regions.

Calculation of these importance weights is the central issue of covariate shift adaptation. The methods for calculating these importance weights will be discussed in Section 7.4. First we have a look at the method used to verify the existence of distribution shift in the data.

7.3.2 Verifying a Distribution Shift

In order to verify the existence of a distribution shift between the training and testing data, we used the Kolmogorov–Smirnov (K-S) test for two samples. It is a non-parametric test that aims to check whether the provided two samples, x^{tr} and x^{te}, are drawn from the same distribution. The K-S statistic quantifies a distance between the empirical distribution function of the two samples. The null hypothesis H_0 is that the two samples are drawn from the same distribution, and the alternative hypothesis H_a is that they come from different distributions and we have a covariate shift.

The K-S statistical test for verifying the covariate shift in the data is generally attractive because it is distribution free. It makes use of each individual data point in the samples. However, this test is only easy to compute for one-dimensional input data.

Since our data is multi-dimensional, and the K-S test only takes one-dimensional input, we cannot simply pass our data to this test. We can assume that each ith feature is independent of the rest of the features. Although, this is a weak assumption but it allows us to apply non-parametric K-S test on each corresponding ith feature in the training and testing data and test whether the corresponding features come from the same distribution or not. If we do not want to make this assumption, we can also use a non-parametric multi-variate maximum mean discrepancy (MMD) test proposed by Gretton *et al.* (2007). This test tries to make the decision by calculating the difference between the empirical means of training and test distributions by mapping the data into high dimensional space. Due to computational complexities of MMD test, we have decided to stick with K-S test.

To apply K-S test on each corresponding feature, we have used *kstest2* function from Matlab Statistical Toolbox. It returns '0' if the test is passed at 5% significance level. If this is the case for all/most of the features, we conclude that the input data comes from the same distribution, otherwise the data contains shift.

7.4 Calculating Importance Weights

In this section, we discuss several methods that have been proposed in the literature for the estimation of importance weights. We look at two types of methods for calculating the importance weights: one which tries to estimate the probability density functions directly and the others which estimate the weights directly from the input data without estimating the densities.

7.4.1 Kernel Density Estimation

One of the naive methods for calculating the importance weights is to directly calculate the probability densities from training and testing data separately and calculate the weights by taking the ratio of the estimated distributions. However direct density estimation is a hard problem especially for large dimensional data. It will suffer from the curse of dimensionality. One would require an infinite amount of data for correctly calculating the distributions which is a severe drawback of this method.

Kernel density estimation (KDE) is a non-parametric method to estimate the probability density function from i.i.d samples. For a Gaussian kernel, KDE can be written as:

$$\hat{p}(x) = \frac{1}{n_{tr}\sqrt{2\pi\sigma^2}} \sum_{i=1}^{n} \boldsymbol{K}(x, x_i) \tag{7.17}$$

where $\boldsymbol{K}$ is the kernel matrix and σ is the kernel width. The performance of KDE depends upon the choice of kernel width parameter σ. Kernel width σ can be optimised by likelihood cross-validation explained in Section 7.4.5.

To calculate the importance weights, Equation 7.16 can be directly used by estimating the probability density functions of training ($\hat{p}_{tr}(x)$) and test ($\hat{p}_{te}(x)$) data by using x_{tr} and x_{te} respectively and then estimating the importance weights by $\hat{\beta}(x) = \hat{p}_{te}(x)/\hat{p}_{tr}(x)$. As the number of input dimensions increases, this method of directly estimating the densities suffers from the curse of dimensionality. This is critical when the available training samples are limited. Therefore, the KDE approach will not be reliable in high-dimension problems. Another method for estimating the probability density functions from data is by calculating corresponding histograms. This method is fast but again suffers from the curse of dimensionality.

Because of these severe drawbacks, methods that can calculate the IW without attempting the distribution estimation are much preferred. In the next sections, we review the following three methods which calculate IW without estimating the distribution of training or test data:

- Kernel mean matching (KMM) (Gretton *et al.*, 2009)
- Unconstrained least square importance fitting (uLISF) (Kanamori *et al.*, 2009),
- Kullback–Leibler importance estimation procedure (KLIEP) (Tsuboi *et al.*, 2008)

7.4.2 Kernel Mean Matching

Gretton *et al.* (2009) have proposed the kernel mean matching (KMM) method which works by minimising the means of importance weighted training and test data distributions in a high-dimensional feature space. The high dimensional space is induced by a kernel function, usually a Gaussian kernel, and the difference between the two distributions is reduced by minimising the difference between corresponding means in the high dimensional space. There is no need for density estimation, which means that there is no requirement for a large amount of input data. This allows us to obtain importance estimates directly at the training input points.

The KMM method is based upon maximum mean discrepancy (MMD) measure introduced in Gretton *et al.* (2007). MMD is a non-parametric distance estimate between two

distributions based upon their means in the kernel induced feature space. The important thing is that for two density functions p and q, $\mathrm{MMD}[\Phi, p, q] = 0$ if and only if $p \equiv q$, where Φ is the mapping function. Gaussian kernels are used for transforming the data into high dimensional space and the objective function (J) is given by the difference of the two empirical means:

$$\begin{aligned} J(\beta) = \quad & \min_{\beta} \left|\left| \tfrac{1}{n_{tr}} \textstyle\sum_{i=1}^{n_{tr}} \beta_i \Phi(x_i^{tr}) - \tfrac{1}{n_{te}} \sum_{i=1}^{n_{te}} \Phi(x_i^{te}) \right|\right|^2 \\ \text{subject to} \qquad & \beta_i \in [0, B] \quad \text{and} \quad \left| \tfrac{1}{n_{tr}} \textstyle\sum_{i=1}^{n_{tr}} \beta_i - 1 \right| \leq \epsilon \end{aligned} \tag{7.18}$$

where $B > 0$ and $\epsilon > 0$ are tuning parameters. The objective function can be expanded as:

$$\frac{1}{n_{tr}^2} \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} - \frac{2}{n_{tr}^2} \boldsymbol{\kappa}^T \boldsymbol{\beta} + \text{const.} \tag{7.19}$$

where $\mathbf{K}_{ij} = k\left(x_i^{tr}, x_j^{tr}\right)$ and $\boldsymbol{\kappa}_i = \frac{n_{tr}}{n_{te}} \sum_{j=1}^{n_{te}} k\left(x_i^{tr}, x_j^{te}\right)$. The quadratic problem to find suitable β becomes:

$$\begin{aligned} \min_{\beta} \quad & \tfrac{1}{2} \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} - \boldsymbol{\kappa}^T \boldsymbol{\beta} \\ \text{subject to} \quad & \beta_i \in [0, B] \text{ and } \left| \textstyle\sum_{i=1}^{n_{tr}} \beta_i - n_{tr} \right| \leq n_{tr} \epsilon \end{aligned} \tag{7.20}$$

Since KMM optimisation is formulated as a convex quadratic programming problem, it leads to a unique global solution. It can be seen from the above that $\boldsymbol{\beta}$ depends only upon input training and the test data. Therefore, there is no requirement for estimating the distributions, which is a huge advantage. This is the reason that KMM is expected to work well even in the high dimensional case. These weights allow us to adjust the training data so that it better matches the test data.

There are three parameters that need to be tuned for the algorithm to work well which are: the Gaussian kernel width σ, upper limit of importance weights B and ϵ. The authors have suggested values of $B = 1000$, and $\epsilon = (\sqrt{n_{tr}} - 1/\sqrt{n_{tr}})$ in their paper. However, as the algorithm directly gives the value of the importance weights at training points, there is no direct method for tuning the parameters. In their paper, the authors have recommended $\sigma = 0.1$. We have tested two different values for σ and the results are detailed later.

Implementation in Matlab

We have used Matlab Optimisation Toolbox for solving the quadratic programming problem to calculate the importance weights $\boldsymbol{\beta}$ using KMM. The general form of a QP

problem is:

$$\min_x \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{f}^T\boldsymbol{x} \tag{7.21}$$

$$\text{subject to: } \begin{cases} \boldsymbol{A}.\boldsymbol{x} & \leq & \boldsymbol{b} \\ \boldsymbol{A}_{eq}.\boldsymbol{x} & = & \boldsymbol{b}_{eq} \\ LB \leq & \boldsymbol{x} & \leq UB \end{cases}$$

One can see the similarity between this general form and Equation 7.21. The pseudo code for calculating the importance weights $\boldsymbol{\beta}$ using KMM is given in Algorithm 7.1.

Algorithm 7.1 Pseudo Code for KMM

Input: $\sigma, \{x_i^{tr}\}_{i=1}^{n_{tr}}$, and $\{x_i^{te}\}_{i=1}^{n_{te}}$
Output: $\boldsymbol{\beta}$
1: $\boldsymbol{K}_{i,j} \leftarrow \Phi(x_i^{tr})$;
2: $\boldsymbol{\kappa}_j \leftarrow \frac{n_{tr}}{n_{te}} \sum_i \Phi(x_i^{te})$
3: $B \leftarrow 1000$
4: $\epsilon \leftarrow (\sqrt{n_{tr}} - 1/\sqrt{n_{tr}})$
5: **Prepare for the optimisation:**
6: $\boldsymbol{H} \leftarrow \boldsymbol{K}$
7: $\boldsymbol{f} \leftarrow \boldsymbol{\kappa}$
8: $\boldsymbol{A} \leftarrow [1; -1]$
9: $\boldsymbol{b} \leftarrow [n_{tr}(\epsilon + 1); n_{tr}(\epsilon - 1)]$
10: $\boldsymbol{LB} \leftarrow 0$
11: $\boldsymbol{UB} \leftarrow B$
12: **Running the optimisation:**
13: $\boldsymbol{\beta} = quadprog(\boldsymbol{H}, \boldsymbol{f}, \boldsymbol{A}, \boldsymbol{b}, Aeq, beq, \boldsymbol{LB}, \boldsymbol{UB})$

7.4.3 Unconstrained Least-Squares Importance Fitting

Kanamori *et al.* (2009) have proposed an unconstrained least-squares importance fitting (uLSIF) method for calculating importance weights. They formulate the IW calculation problem as a least-squares function fitting problem. They further obtained a closed form for this method, which means that we do not have to perform a lengthy optimisation, rather the importance weights can be computed efficiently by solving a system of linear equations.

Let us estimate the importance weights $\hat{\beta}(x)$ by a linear model:

$$\hat{\beta}(x) = \langle \boldsymbol{\alpha}, \psi(x) \rangle \tag{7.22}$$

$$= \sum_{l=1}^{b} \alpha_l \psi_l(x) \tag{7.23}$$

where $\alpha_l \in \mathbb{R}$ and $\alpha_l \geq 0$, are the model parameters that we are going to estimate from the input data samples, and $\psi(x)$ are the basis functions. The squared loss $J(\alpha)$ is

defined as the expectation under the probability of training samples. The parameters α in the model $\hat{\beta}(x)$ are estimated so that the following squared loss J between the actual and estimated weights is minimised:

$$J(\alpha) = \frac{1}{2}\int(\hat{\beta}(x) - \beta(x))^2 p_{tr}(x)\, dx \tag{7.24}$$

Expanding and using the empirical averages in the above equations:

$$\begin{aligned}
\hat{J}(\alpha) &= \frac{1}{2n_{tr}}\sum_{i=1}^{n_{tr}}\hat{\beta}(x_i^{tr})^2 - \frac{1}{n_{te}}\sum_{j=1}^{n_{te}}\hat{\beta}(x_j^{te})^2 + \text{const.} \\
&= \frac{1}{2}\sum_{l,l'=1}^{b}\alpha_l\alpha_{l'}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\psi_l(x_i^{tr})\psi_{l'}(x_i^{tr})\right) - \sum_{l=1}^{b}\alpha_l\left(\frac{1}{n_{te}}\sum_{i=1}^{n_{te}}\psi_l(x_i^{te})\right) \\
&= \frac{1}{2}\boldsymbol{\alpha}^T\hat{\boldsymbol{H}}\boldsymbol{\alpha} - \hat{\boldsymbol{h}}^T\boldsymbol{\alpha}
\end{aligned} \tag{7.25}$$

where $\hat{\boldsymbol{H}}$ is a $b \times b$ matrix with each element given by:

$$\hat{\boldsymbol{H}}_{l,l'} = \frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\psi_l(x_i^{tr})\psi_{l'}(x_i^{tr}) \tag{7.26}$$

and $\hat{\boldsymbol{h}}$ is a b-dimensional vector with each element given by:

$$\hat{\boldsymbol{h}}_l = \frac{1}{n_{te}}\sum_{i=1}^{n_{te}}\psi_l(x_i^{te}) \tag{7.27}$$

Taking into account the non-negativity constraint on the weights $\beta(x)$, we can formulate our optimisation problem as follows:

$$\begin{aligned}
\min_{\alpha} \quad & \tfrac{1}{2}\boldsymbol{\alpha}^T\hat{\boldsymbol{H}}\boldsymbol{\alpha} - \hat{\boldsymbol{h}}^T\boldsymbol{\alpha} \\
\text{subject to} \quad & \boldsymbol{\alpha} \geq \mathbf{0}
\end{aligned} \tag{7.28}$$

where $\mathbf{0}$ is a b-dimensional vector with all zeros. This is a quadratic optimisation problem which can be solved very easily using Matlab Optimisation Toolbox or any other similar toolbox.

To obtain unconstrained formulation of the above problem, the authors ignore the non-negativity constraint and introduce a regularisation term in the optimisation. This results in the following unconstrained optimisation problem:

$$\min_{\alpha\in\mathbb{R}^b} \quad \tfrac{1}{2}\boldsymbol{\alpha}^T\hat{\boldsymbol{H}}\boldsymbol{\alpha} - \hat{\boldsymbol{h}}^T\boldsymbol{\alpha} + \tfrac{1}{2}\lambda\boldsymbol{\alpha}^T\boldsymbol{\alpha} \tag{7.29}$$

The above optimisation problem is an unconstrained quadratic optimisation problem which can be computed as:

$$\widetilde{\boldsymbol{\alpha}} = \max(0, (\boldsymbol{H} + \lambda \boldsymbol{I}_b)^{-1} \hat{\boldsymbol{h}}) \tag{7.30}$$

where $\boldsymbol{I}_b$ is a b-dimensional identity matrix.

The advantage of the above unconstrained least square importance fitting (uLSIF) is that the solution can be computed quickly by solving a system of linear equations. Therefore its computation is much faster than solving a quadratic optimisation problem. As the optimisation returns the parameters of the model, out of sample prediction is also possible. This property is very helpful for tuning the parameters of the optimisation. For the basis function, Gaussian kernel with width σ is used which is chosen by likelihood cross validation as explained in Section 7.4.5.

7.4.4 Kullback–Leibler Importance Estimation Procedure

The Kullback–Leibler importance estimation procedure (KLIEP) proposed by Tsuboi *et al.* (2008) uses divergence between the importance based estimated test distribution to the true test distribution in terms of Kullback-Leibler (KL) divergence. It uses the special property of KL divergence between two probability distributions P and Q that $KL(P||Q) = 0$ if and only if $P \equiv Q$ otherwise it is always greater than 0.

Let us formally look at the mathematical formulation of this method. Let us estimate $\hat{\beta}(x)$ by a linear model:

$$\hat{\beta}(x) = \langle \boldsymbol{\alpha}, \psi(x) \rangle$$

where $\alpha_l \in \mathbb{R}$ and $\alpha_l \geq 0$ are the model parameters that we are going to estimate and $\psi(x)$ is the linear basis function. Using this estimated importance $\hat{\beta}(x)$ we can estimate the test distribution $\hat{p}_{te}$ as

$$\hat{p}_{te}(x) = p_{tr}(x)\hat{\beta}(x)$$

Now the parameter α can be learned so that the Kullback-Leibler divergence from $p_{te}(x)$ to $\hat{p}_{te}(x)$ is minimised:

$$\begin{aligned} KL[p_{te}(x)||\hat{p}_{te}(x)] &= \int_{n_{te}} p_{te}(x) \log \frac{p_{te}(x)}{\hat{p}_{te}(x)} dx \\ &= \int_{n_{te}} p_{te}(x) \log \frac{p_{te}(x)}{p_{tr}(x)} dx - \int_{n_{te}} p_{te}(x) \log \hat{\beta}(x) dx \end{aligned} \tag{7.31}$$

Since the first term in Equation 7.31 is independent of α, we can ignore it and concentrate on maximising the second term which is:

$$\int_{n_{te}} p_{te}(x) \log \hat{\beta}(x) dx \approx \frac{1}{n_{te}} \sum_{x \in n_{te}} \log \hat{\beta}(x)$$

This is the objective function which is to be maximised. The optimisation problem is expressed as:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \textstyle\sum_{x \in n_{te}} \log\langle \boldsymbol{\alpha}, \psi(x)\rangle \\ \text{subject to} \quad & \textstyle\sum_{x \in n_{tr}} \langle \boldsymbol{\alpha}, \psi(x)\rangle = n_{tr} \text{ and} \\ & \alpha_l \geq 0 \end{aligned} \tag{7.32}$$

The global solution can be obtained by performing gradient ascent iteratively on this objective function. As the basis function, $\psi(x)$, the authors have suggested to use Gaussian kernels and the kernel parameter σ is selected by applying validation on the test data. Pseudo code for calculating importance weights using the KLIEP method is given in Algorithm 7.2.

Algorithm 7.2 Pseudo Code for KLIEP

Input: $m = \{\psi_l(x)\}_{l=1}^{b}, \{x_i^{tr}\}_{i=1}^{n_{tr}}$, and $\{x_i^{te}\}_{i=1}^{n_{te}}$
Output: $\boldsymbol{\alpha}, score$
  $A_{j,l} \leftarrow \psi_l(x_j^{te})$;
  $b_l \leftarrow \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \psi_l(x_i^{tr})$
  $c = \boldsymbol{b}^T \boldsymbol{b}$
  Initialise $\boldsymbol{\alpha}(> \mathbf{0})$, $score \leftarrow 0$, and $\varepsilon(0 < \varepsilon \ll 1)$;
  repeat
    $\hat{\boldsymbol{\alpha}} \leftarrow \hat{\boldsymbol{\alpha}} + \varepsilon \boldsymbol{A}^T (1./\boldsymbol{A}\hat{\boldsymbol{\alpha}})$;
    $\hat{\boldsymbol{\alpha}} \leftarrow \hat{\boldsymbol{\alpha}} + (1 - \boldsymbol{b}^T \hat{\boldsymbol{\alpha}}) \boldsymbol{b} / \boldsymbol{b}^T \boldsymbol{b}$;
    $\hat{\boldsymbol{\alpha}} \leftarrow \max(\mathbf{0}, \hat{\boldsymbol{\alpha}})$;
    $scôre \leftarrow mean \log(A\alpha)$
    if $scôre - score < 0$ **then**
      $converge \leftarrow 1$
    else
      Update:
      $\boldsymbol{\alpha} \leftarrow \hat{\boldsymbol{\alpha}}$
      $score \leftarrow scôre$
    end if
  until $converge == 1$
  $\hat{\boldsymbol{\beta}}(x) \leftarrow \sum_{l=1}^{b} \alpha_l \psi_l(x)$

The biggest advantage of using this method is that it relies on the testing not on training data to estimate all optimisation parameters. This is very useful in the scenarios when a large amount of testing data is available as compared to the training data. There is only one parameter which is the kernel width σ that needs to be tuned. Applying cross validation for tuning the parameters should not be a problem, as usually testing data is available in abundance. It is selected by applying likelihood cross validation explained in Section 7.4.5.

7.4.5 Likelihood Cross Validation

Sugiyama *et al.* (2006) presented a cross validation scheme that can be applied for tuning the parameters on the input data which has covariate shift in it. This method is slightly different from standard validation methods as cross validation is applied on training as well as testing data simultaneously.

It is done as follows: First divide training samples $\{x_i^{tr}\}_{i=1}^{n_{tr}}$ and testing samples $\{x_j^{te}\}_{j=1}^{n_{te}}$ into R disjoint subsets $\{X_r^{tr}\}_{r=1}^{R}$ and $\{X_r^{te}\}_{r=1}^{R}$, respectively. Then an importance estimate $\hat{\beta}_{X_r^{tr},X_r^{te}}(x)$ is obtained using $\{X_j^{tr}\}_{j\neq r}$ and $\{X_j^{te}\}_{j\neq r}$, i.e., without X_r^{tr} and X_r^{te}, and the cost J is approximated using the held out samples X_r^{tr} and X_r^{te}. This procedure is repeated for $r = 1, 2, \ldots, R$ and its average over R trials $\hat{J}^{CV}$ is used as an estimate of J. The value of σ which maximises $\hat{J}^{CV}$ is selected as the optimal value.

Let us consider the example of KLIEP method for which the objective function is given in Equation 7.32. The test samples $\{x_j^{te}\}_{j=1}^{n_{te}}$ are divided into R disjoint subsets $\{X_r^{te}\}_{r=1}^{R}$. Then the importance estimate $\hat{\beta}_r(x)$ is obtained from $\{X_j^{te}\}_{j\neq r}$ and the score $\hat{J}_r$ is approximated using X_r^{te} as

$$\hat{J}_r = \frac{1}{|X_r^{te}|} \sum_{x \in X_r^{te}} \log \hat{\beta}_r(x) \tag{7.33}$$

This procedure is repeated for $r = 1, 2, \ldots, R$ and its average over R trials $\hat{J}^{CV}$ is used as an estimate of J as:

$$\hat{J}^{CV} = \frac{1}{R} \sum_{R} \hat{J}_r \tag{7.34}$$

The value of σ which maximises $\hat{J}^{CV}$ is selected as the optimal value. Pseudo code for the LCV procedure for KLIEP is given in Algorithm 7.3.

Algorithm 7.3 Pseudo Code for KLIEP model selection by LCV

Input: $M = \{m_k | m_k = \{\psi_l^{(k)}(x)\}_{l=1}^{b^{(k)}}, \{x_i^{tr}\}_{i=1}^{n_{tr}}, \text{ and } \{x_i^{te}\}_{i=1}^{n_{te}}$
Output: $\hat{\beta}(x)$
  Split $\{x_j^{te}\}_{j=1}^{te}$ into R disjoint subsets $\{X_r^{te}\}_{r=1}^{R}$;
  for each model $m \in M$ **do**
    for each split $r = 1, 2, \ldots, R$ **do**
      $\hat{\beta}_r(x) \leftarrow KLIEP(m, \{x_i^{tr}\}_{i=1}^{n_{tr}}, \{X_j^{te}\}_{j\neq r})$;
      $\hat{J}_r(m) \leftarrow \frac{1}{|X_r^{te}|} \sum_{x \in X_r^{te}} \log \hat{\beta}_r(x)$;
    end for
    $\hat{J}(m) \leftarrow \frac{1}{R} \sum_R \hat{J}_r(m)$;
  end for
  $\hat{m} \leftarrow \text{argmax}_{m \in M} \hat{J}(m)$;
  $\hat{\beta}(x) \leftarrow KLIEP(m, \{x_i^{tr}\}_{i=1}^{n_{tr}}, \{x_j^{te}\}_{j=1}^{n_{te}})$;

Table 7.2: Details of the methods used for importance weighting.

Method	Density estimation	Model selection	Optimisation	Out-of-sample prediction
KDE	Required	Available	Analytical	Possible
KMM	Not required	Not available	Convex QP	Not Possible
uLSIF	Not required	Available	Analytical	Possible
KLIEP	Not required	Available	Convex non-linear	Possible

7.4.6 Differences between KDE, KMM, uLSIF and KLIEP

KDE is computationally efficient as no optimisation is involved. However, it does not always estimate the distributions correctly as it suffers from the curse of dimensionality. In contrast, KMM tries to overcome the curse of dimensionality by directly estimating the importance weights. It gives the IW values directly at the training points and currently there is no model selection method that can be applied to tune the parameters. Therefore, model parameters such as Gaussian width σ need to be determined by prior knowledge of the domain. Furthermore, the computation of KMM is expensive as a QP problem has to be solved.

Other two methods, KLIEP and uLSIF, also do not require density estimation for calculating the importance weights. They give an estimate of the parameters of the importance function as compared to importance values given by KMM. Therefore, the values of the importance weights at unseen points can be estimated by KLIEP and uLSIF. This feature is very useful as it enables us to apply LCV for model selection which is a significant improvement over KMM.

However, KLEIP has to solve a convex non-linear optimisation problem which is why it is still computationally expensive compared to uLSIF. The uLSIF method has a closed form and the solution can be computed in an efficient manner by solving a system of linear equations. One advantage that KMM has over KLIEP and uLSIF is that all the parameters of the optimisation are preselected. Therefore, there is no need for any parameter selection and optimisation process. This parameter optimisation step does add to the computational costs of KLIEP and uLSIF as we are going to observe in Section 7.6.5.

7.5 Use of IW for Classification

After calculating the importance weights (IW), the next step is to incorporate them into a chosen supervised machine learning algorithm. In this section, we derive the formulations to incorporate IW into two of the most commonly used machine learning

algorithms, logistic regression (IW-LR) and support vector machines (IW-SVM), so that the classifier can compensate for the covariate shift in the data.

7.5.1 Importance Weighted Logistic Regression

A standard logistic regression uses the following linear model:

$$f_\theta(x) = \langle \theta, \phi(x) \rangle \tag{7.35}$$

where θ_i are parameters parametrising the space of linear functions and $\phi(x)$ is a basis function of x. Here we consider binary classification, i.e., $y = \{-1, +1\}$. Let us model the posterior probability of class y given x using the sigmoid function as:

$$p_\theta(y|x) = 1/(1 + \exp(-y f_\theta(x))) \tag{7.36}$$

The optimum value of θ can be selected by maximising ordinary logistic regression function as:

$$\hat{\boldsymbol{\theta}}_{LR} = \underset{\theta}{\operatorname{argmax}} \left[\sum_{n_{tr}} (y f_\theta(x) - \log(1 + \exp(y f_\theta(x)))) \right] \tag{7.37}$$

or we can minimise the negative of the above equation. This method will be suitable for the ordinary case where there is no shift in the data but it will not be consistent under covariate shift. To compensate for the covariate shift, the calculated importance weights (IW) are incorporated in the learning method by taking the weighted sum as follows:

$$\hat{\boldsymbol{\theta}}_{LR} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[\sum_{n_{tr}} \beta(x)(\log(1 + \exp(y f_\theta(x))) - y f_\theta(x)) \right] \tag{7.38}$$

This type of logistic regression is called importance weighted logistic regression (IW-LR). The above optimisation problem can be easily solved by using any gradient descent method.

7.5.2 Importance Weighted Support Vector Machines

In importance weighted SVMs (IW-SVMs) what we want to do is to shift the separating hyperplane in a way to take into consideration the more important data which gives better representation of the test data. For doing this, we introduce the importance weights β_i into the SVM optimisation function. Instead of using a fixed penalty for all of the training data, IWs are used to increase the penalty for the data points that are more important and reduce it for less important data points. The quadratic optimisation problem becomes:

$$\min_w \frac{1}{2} |w|^2 + C \sum_{i=1}^{L} \beta_i \xi_i \tag{7.39}$$

subject to:

$$\begin{aligned} y_i(\langle w, x_i\rangle + b) &\geq 1 - \xi_i \\ 1 \leq i &\leq L \\ \xi_i &\geq 0 \end{aligned}$$

The effect of this function is that the separating hyperplane is adjusted to consider the important data according to the importance weights β_i. The dual of the above optimisation becomes:

$$L_D(w, b, \alpha, \xi, \beta) \equiv \frac{1}{2}|w|^2 + C\sum_{i=1}^{L}\beta_i\xi_i - \sum_{i=1}^{L}\alpha_i(y_i(\langle w, x_i\rangle + b) - 1 + \xi_i) \quad (7.40)$$

Differentiating the above equation w.r.t. w, b and ξ and setting the derivatives to zero, we get the following optimisation problem:

$$\begin{aligned} \frac{\partial L_D}{\partial w} = 0, &\quad \Rightarrow \quad w = \sum_i \alpha_i y_i x_i \\ \frac{\partial L_D}{\partial b} = 0, &\quad \Rightarrow \quad \sum_i \alpha_i y_i = 0 \\ \frac{\partial L_D}{\partial \xi_i} = 0, &\quad \Rightarrow \quad C = \frac{\alpha_i}{\beta_i} \end{aligned}$$

The final IW-SVM optimisation problem looks like:

$$\max_{\alpha} \sum_{i=1}^{L}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} \quad \text{s.t.} \quad 0 \leq \alpha_i \leq \beta_i C \quad \text{and} \quad \sum_{i=1}^{L}\alpha_i y_i = 0 \quad (7.41)$$

This IW-SVM is going to adapt the separating hyperplane to consider the weights calculated from the training and testing data. This type of SVM should perform better than a standard SVM if there exists a data shift in the input data. However, it might reduce the performance of the classifier if there is no shift in the data. We have used the SVM toolbox for Matlab by S. Gunn for this work. This toolbox is available at http://www.isis.ecs.soton.ac.uk/resources/svminfo/. We have modified the original code so that it can consider the importance weights as given by Equation 7.41. The Matlab code for IW-SVM has been made available for public use at http://eprints.soton.ac.uk/337383/2/supp_matlabCode.zip.

7.6 Illustration on Example Toy Data

In this section, we test and report the results of applying importance weighting methods explained in Section 7.4 on an artificially generated toy dataset which has induced data

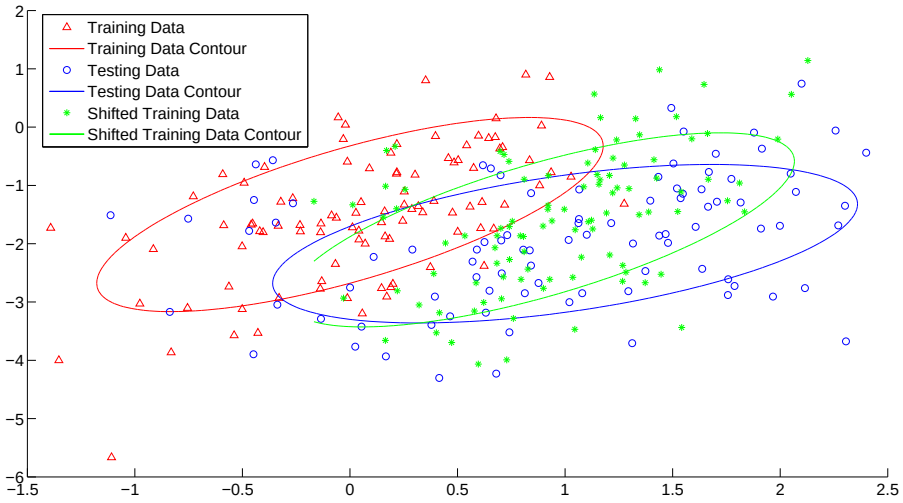


Figure 7.3: Scatter plot showing how the training data has shifted towards the testing data by using KMM on a toy dataset. Means and covariance of training and testing data are: $\mu_{tr} = [0, -1.5]$, $\mu_{te} = [1, -2]$, $\Sigma_{tr} = [0.5\ 0.5; 0\ 1]$ and $\Sigma_{te} = [1\ 0.5; 0\ 1]$ respectively. A Gaussian kernel was used with $\sigma = 1$ for the shifted data.

shift to illustrate their effects. For the classification task, modified SVMs have been used which consider importance weights while estimating the separating hyperplanes.

7.6.1 Testing on 2D One Class Data

As an illustration of how the training data will shift after applying importance weighting, we apply KMM method to a one class toy problem generated by using two 2D-Gaussian distributions. In Figure 7.3, 100 training and 100 test samples have been plotted along with the KMM importance weighted data. The training and testing data have been generated to artificially induce the data shift between the two by using different means ($\boldsymbol{\mu}$) and covariance ($\boldsymbol{\Sigma}$) matrix.

To verify the existence of covariate shift we applied K-S test explained in Section 7.3.2 on this toy data. First we apply the test on two dimensional data by considering each corresponding feature independently from training and testing data. For both features, the K-S test failed, returning '1' for each, certifying that both corresponding features come from two different distributions.

To compensate for covariate shift in this dataset, KMM importance weighting was applied on the input data to obtain weights (β) with $\sigma = 1$ and the new mean of the shifted data was calculated by multiplying each training point x_i with corresponding

Table 7.3: Details of the parameters used for illustrative 2D classification data taken from Tsuboi *et al.* (2008).

	Training		**Testing**	
	$y=0$	$y=1$	$y=0$	$y=1$
μ	$(-1,0)$	$(4,2)$	$(0,2)$	$(3,1)$
Σ	$\begin{pmatrix} 0.75 & 0 \\ 0 & 1.5 \end{pmatrix}$		$\begin{pmatrix} 0.75 & 0 \\ 0 & 1.5 \end{pmatrix}$	

weights (β_i):

$$\mu_{shift} = \frac{1}{n_{tr}} \sum_{i=0}^{ntr} \beta_i x_i$$

The new mean ($\boldsymbol{\mu}_{shift}$) was used to plot the contour of the shifted data which is marked as green coloured '*' in the figure. It can be seen from the plots that the training data has clearly shifted towards the testing data. To verify if importance weighting has compensated for the shift in data, we apply the K-S test again on the shifted training and test data. As expected, both features pass the test returning '0' for each one of them, verifying that KMM has been able to successfully compensate for the shift in the input data.

7.6.2 Testing on 2D Two Class Data

In this section we detail the classification experiments on two class toy data generated using Gaussian distributions. The input data consists of 200 training and testing points each. The details of each distribution are given in Table 7.3 which have been taken from Tsuboi *et al.* (2008). We apply all four methods, i.e., KDE, KMM, KLIEP and uLSIF, for estimating the importance weights on this artificial data and use importance weighted SVMs for the classification.

Figure 7.4 shows the scatter plot of two class toy data generated by Gaussian distributions with the specifications given in Table 7.3. The data is generated with specific mean and variance to artificially induce shift to see the effectiveness of IW methods. The decision boundary for separating the two classes is plotted after applying KDE, KMM, KLIEP, uLSIF and no importance weighting on the testing and training data using linear-SVMs with fixed $C = 0.1$.

It is observed that the decision boundary based upon KDE has slightly deviated from the standard decision boundary shown by solid line generated by standard SVMs without any reweighting. One of the main reasons for this is small amount of input data available for training. As we don't have very large testing and training datasets, the estimates for the probability distributions are not very precise which is why the decision boundary is not shifted considerably towards the testing data and the calculated weights are close to 1. We will get better estimates of the distributions if we had more input data.

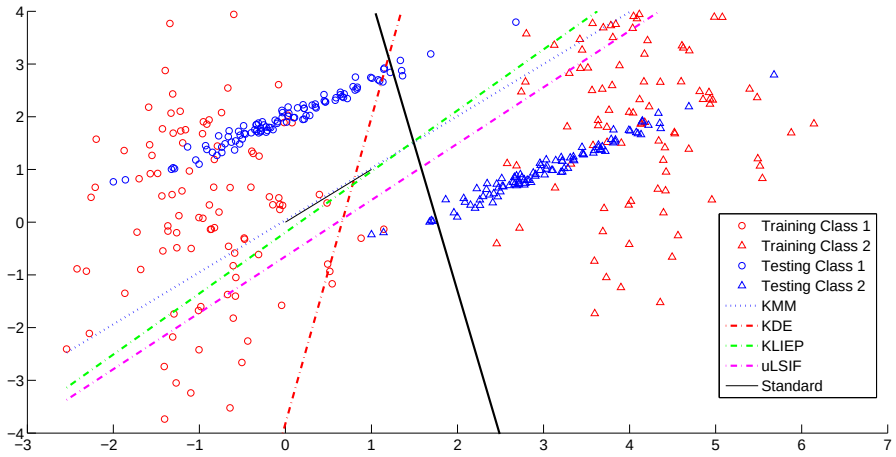


Figure 7.4: Scatter plot showing the decision boundary of binary classification with SVM using: KDE, KMM, KLIEP, uLSIF and standard SVM, on the 2D toy data.

In comparison to KDE, KMM with $\sigma = 0.1$, KLIEP, and uLSIF give much better decision boundaries on the shifted test data, which proves that algorithms which directly compute the importance weights without calculating the input densities can work reasonably well even on small amounts of input data.

There are three interesting observations which need further investigation:

- What is the effect of optimising the SVM classifier by varying the value of C. The value of C is of particular interest as it controls the trade-off between test errors and smoothness of the decision boundary which is what we wish to control using importance weighting.
- What is the effect of amount of input data available for training? A larger amount should work in favour of KDE for the reasons explained earlier. However, its effect on the other algorithms for calculating IW is also very important.
- The third thing that needs to be investigated is the computational time taken for the processing all of these algorithms.

In the next sections, we investigate these questions in detail.

7.6.3 Effect of Varying C

For the first experiments, we have used linear kernel SVMs with varying values of C. It is varied over the range $C \in \{0.1, 1, 10, 100, 1000, 10000\}$ with fixed amount of training and testing data. Standard deviation is obtained by running the tests 10 times. For each

Table 7.4: Classification performance on 2D toy data by varying C. Results are presented as average percentage error with standard deviation over 10 trials each.

Method	**Value of C**					
	0.1	1	10	100	1000	10000
Standard SVM	8.8 (2.6)	8.7 (4.3)	10.3 (3.4)	8.3 (3.6)	10.3 (4.7)	9.8 (7.0)
KDE	16.3 (19.1)	8.1 (3.7)	10.1 (3.1)	8.2 (3.5)	10.3 (4.7)	9.7 (7.0)
KMM-$\sigma = 0.1$	1.9 (2.2)	2.3 (2.1)	10.3 (7.2)	11.3 (4.9)	13.7 (7.4)	16.9 (10.9)
KMM-$\sigma = 1.0$	35.1 (28.1)	8.8 (5.4)	10.7 (2.9)	8.5 (3.9)	10.7 (6.4)	10.9 (9.6)
KLIEP	8.0 (8.6)	4.6 (5.8)	8.8 (9.6)	4.9 (4.5)	9.8 (8.7)	9.5 (10.1)
uLSIF	0.9 (1.0)	2.2 (3.7)	4.1 (4.3)	6.2 (5.5)	5.8 (6.3)	8.5 (5.8)

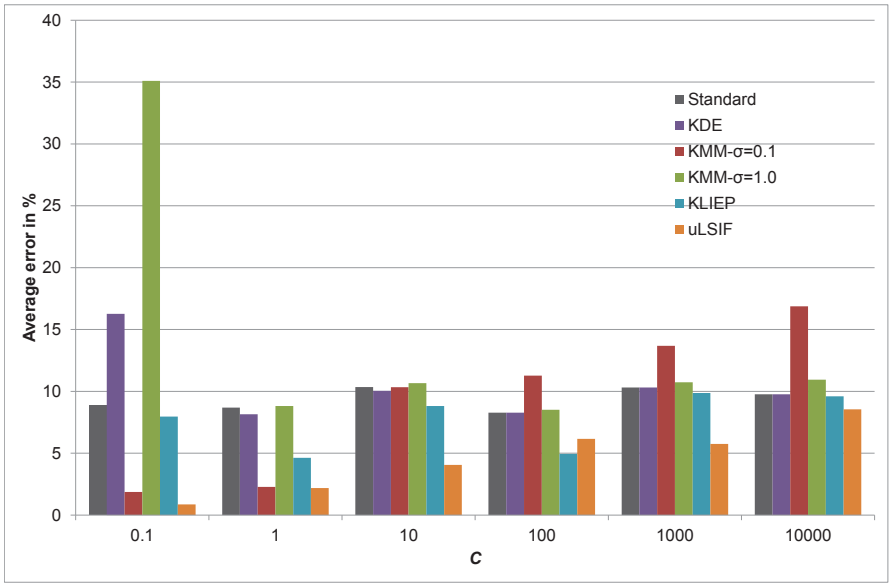


Figure 7.5: Classification performance on 2D toy data with varying value of C. Results are presented as average percentage error with standard deviation over 10 trials each.

iteration, 100 training and 10,000 testing samples were generated using 2D Gaussians and the parameters given in Table 7.3. Table 7.4 shows the results of applying standard SVM with no IW, KDE, KMM, KLIEP and uLSIF weighted SVMs. For all of the tests, linear-SVMs have been used. For KMM, we have used two different values of the kernel width, $\sigma \in \{0.1, 1.0\}$, as there is no direct method for optimising the parameters for this algorithm. In comparison, KLIEP and uLSIF use the likelihood cross validation method explained in Section 7.4.5 for selecting the optimal values of the corresponding parameters.

Figure 7.5 shows the bar plots of the data in Table 7.4. A few interesting observations can be made: KMM and uLSIF-based reweighting method work very well for the small value of C while their performance degrades for a larger value of C. This means that these methods perform well for the classifiers which put high priority on a smooth decision

boundary and are not very well tuned to cater for the true input data complexity. The performance gain by using IW-SVM is such that it gives better results than a highly tuned standard SVM. However if the classifier is already complex enough to capture the fine details of the input data, i.e., the classifier parameters have been tuned for the input data, IW may not improve the performance rather it may degrade it.

This is observed from the figure, where we see that there is no real performance gain of applying IW for large values of C. In some of the cases, the error rate becomes worse than the standard SVMs. Therefore, one can deduce that for a highly tuned classifier, IW the training data might reduce the classification performance. However, in the case of a classifier which puts more weight on the generality of data rather than completely fitting the training data, IW will improve the results. There can be several situations where tuning parameters of the classifier can be a bit difficult, e.g., when we have a very large amount of data and tuning of the parameters is very costly or is time constrained. In these situations, an appropriate IW method will definitely improve the results.

Out of the two tests in which we have used IW-KMM with $\sigma \in \{0.1, 1.0\}$, KMM with $\sigma = 1.0$ performs poorly for the small value of C. However, for larger values of C, it does perform better than KMM with $\sigma = 0.1$. This means that the selection of the kernel width is very important in KMM based IW. Methods for selecting a proper value of σ for KMM is still an open research question.

Generally speaking, best results are obtained by using uLSIF which gives best results in five out of six tests. However, KLIEP also performs consistently better than the standard SVM for all values of C. One of the main reason for this is that KLIEP has an inbuilt method for selecting the appropriate width of the kernel which is not present in KMM. Because of this, it is able to optimise for the particular problem and perform consistently better. The KDE based re-weighting does outperform the standard SVM in four out of six tests, however the performance gain is not big.

7.6.4 Effect of Training Data

The second thing that we would like to test is the effect of available training data for calculating the importance weights. We expect that KMM and KDE will be affected by this while the KLIEP and uLSIF should not be affected as they do not use training data for calculating the importance weights.

For this experiment, we have again used linear-SVMs with a fixed value of $C = 0.1$. The training and testing data is generated by using 2D Gaussians with the parameters given in Table 7.3. Each experiment is run 10 times using different size of training data while testing on 10,000 samples. The results of running these experiments on the toy data with varying values of n_{tr} are shown in Table 7.5.

Table 7.5: Classification performance on 2D toy data for varying size of the training data (n_{tr}). Results are presented as average percentage error with standard deviation over 10 trials each.

Method	**Value of n_{tr}**				
	100	300	500	700	900
Standard SVM	8.9 (2.6)	8.4 (1.2)	8.2 (1.2)	8.3 (1.2)	8.5 (1.6)
KDE	16.3 (19.2)	21.5 (23.9)	14.5 (2.4)	10.1 (6.3)	9.1 (4.7)
KMM-$\sigma = 0.1$	1.9 (2.2)	2.7 (3.1)	0.8 (1.1)	3.7 (5.3)	2.1 (2.2)
KMM-$\sigma = 1.0$	35.1 (28.1)	37.9 (25.8)	42.8 (23.2)	45.1 (27.2)	30.6 (33.7)
KLIEP	8.0 (8.6)	6.0 (7.6)	2.9 (3.5)	3.6 (3.7)	4.7 (2.3)
uLSIF	0.9 (1.0)	1.9 (1.8)	0.7 (1.1)	5.1 (7.8)	3.3 (4.0)

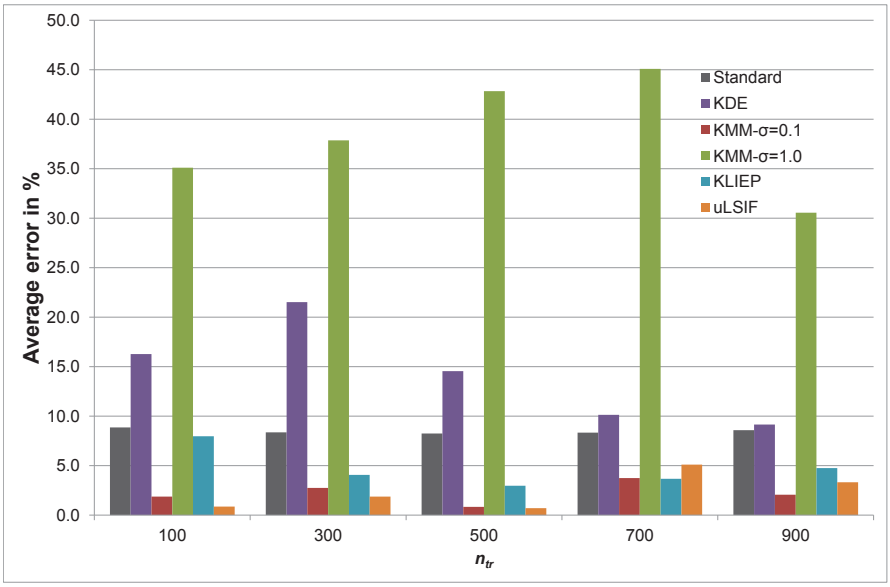


Figure 7.6: Classification performance on 2D toy data with varying size of training data (n_{tr}). Results are presented as average percentage error with standard deviation over 10 trials each.

Figure 7.6 shows the bar plots of the data shown in Table 7.5. It has been already observed that KMM with $\sigma = 1.0$ does not perform well for small values of C. However, KMM with $\sigma = 0.1$ performs as well as uLSIF. For the case of KDE, as expected, this method does show clear improvement in the average error results with the increase in the training data. As n_{tr} increases, the algorithm has more data to better estimate the probability densities, hence the calculated weights are more accurate which is reflected in the final accuracies. As KLIEP does not use training data for calculating the importance weights, varying size of the training data does not have as such any effect on it and remains pretty much constant for all sizes of the training data.

7.6.5 Computational Costs of the Algorithms

The last thing which we wish to test is the computational costs for running all of these IW algorithms. As we have already seen, KDE does not perform really well when there is not much available input data. Therefore, for these tests we ignore this method and concentrate on the other three methods namely: KMM, KLIEP and uLSIF.

Intuitively, uLSIF should be the fastest as all of the parameters of this algorithm can be calculated easily by solving a system of linear equations as mentioned earlier. KLIEP should be faster than KMM similar to uLSIF but it does involve a step of cross validation for searching for optimal parameters for the algorithm. For KMM all of the parameters are fixed as there is no direct way for searching for the optimal parameters. This could be a drawback for KLIEP and uLSIF in terms of computational time as we calculate the time for the complete process.

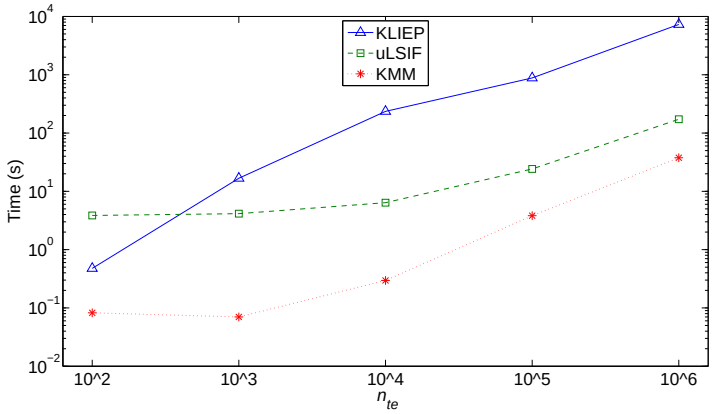
The number of training input points has been fixed to $n_{tr} = 100$, while varying the dimensions $d = 10, 100, 1000$ of the data. The number of test samples is varied over the range $n_{te} = 10^2, 10^3, \ldots, 10^6$. Each experiment for different value of d and n_{tr} is repeated 10 times on LEDA3 which is a server with dual core Intel® Xeon™CPU 2.60 GHz with 48GB of RAM running Linux in the CSPC research group, University of Southampton.

Figure 7.7 shows the computational time in seconds for calculating the importance weights using these three algorithms. KLIEP does perform well for small dimensions and number of the testing data points. However, KMM and uLSIF are much better for high dimensional and large testing datasets. For all of these algorithms, the computation time increases with the increase in the size of the training data. Out of the three algorithms tested, KMM is the fastest. The only reason for this could be the lack of parameter selection step which slows down the other two methods. Out of KLIEP and uLSIF, the latter is the quicker as the solution is obtained by solving a system of linear equations rather than solving an optimisation problem.

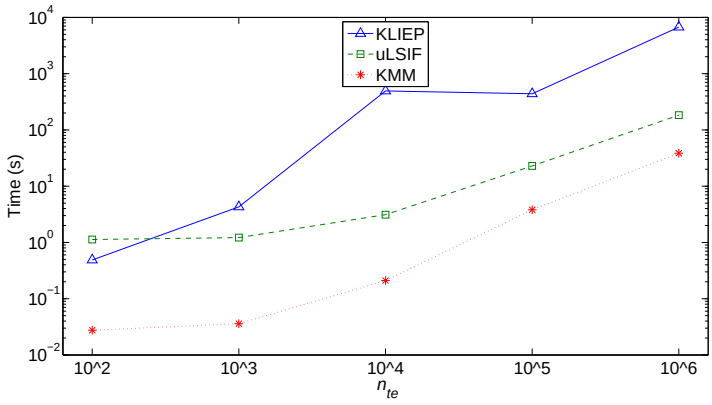
After testing the importance weighting algorithms on toy data, in the next section we will give the results of applying these to compensate for speaker and acoustic environment differences for intra– and inter–database emotion classification.

7.7 Evaluation Setup & Results for Emotion Classification

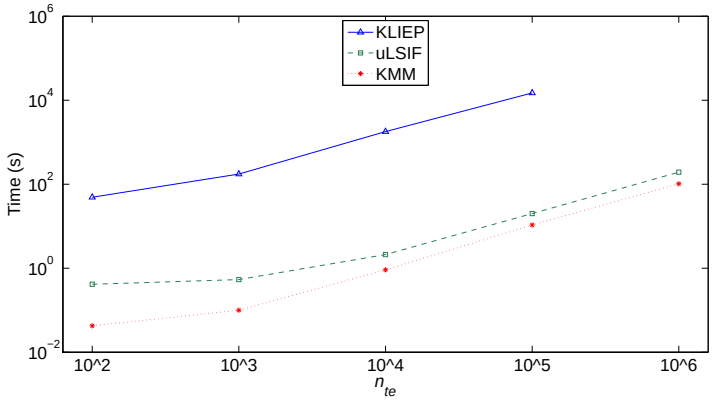
It was mentioned previously that CMN and MLLR (CMN+MLLR) are the algorithms of choice in traditional ASR systems to compensate for different speakers and acoustic environments. In this section, we will test how these algorithms perform for emotion recognition in intra– and inter–database emotion classification. These tests will show if CMN+MLLR has any positive effect on the classification accuracy by compensating



(a) $d = 10$



(b) $d = 100$



(c) $d = 1000$

Figure 7.7: Average log time in seconds taken for calculating the importance weights over 10 trails for varying input testing data (n_{te}) size. The x-axis shows the number of testing points (n_{te}) and the y-axis shows the average log-time in seconds.

for the speaker and environment differences. We compare these results with that of IW-algorithms (KMM, KLIEP, uLSIF) used in IW-SVMs for the same task to see their effectiveness on individual databases. For intra-database classification setup, we have used the first 3000 features selected by Borda-hard feature ranking method for each corresponding database.

In the second setup, we perform inter–database emotion classification which exaggerates the acoustic and speaker differences between the training and testing databases. It also adds another difference of language between the two. For inter–database classification, one of the databases is used for testing while the rest are used for training the classifiers. This procedure can be described as leave-one-database-out cross validation. The three acted speech databases (DES, Berlin and Serbian) contain speech samples from adult speakers, while the Aibo database contains data from children. Combining these two separate groups of databases will add another major difference of mixing adult speakers with children which is why we have decided to keep them separate. This also allows us to compare our results against those of the Interspeech 2009 Emotion Challenge in which Aibo-Ohm was used for training while Aibo-Mont for testing. For all of these inter–database emotion classification tests, we have used the universal feature set (Uni-Large) consisting of over 5000 features explained in Section 5.6.

7.7.1 Mapping of Emotion Classes

It has been discussed before that each emotional speech database has a different number of classes per database. Hence, we can not directly apply inter–database classification. The three acted emotional speech database (DES, Berlin and Serbian) have four emotion classes common among each other. These classes are *neutral*, *angry*, *happy* and *sad* as was shown in Table 3.1. The spontaneous speech databases (Aibo-Mont and Aibo-Ohm) have all five classes common among each other. We can apply inter–database emotion classification on these common classes.

Another solution is to map all classes on a lower dimensional space. For doing this mapping, *valence* and *arousal* dimensions are our best options. It has been discussed in Chapter 6 that *valence* is not very obvious from the data-driven NMDS plots. However, testing on these dimensions individually will give us further insight into their presence in the data. We expect that classification accuracy for *valence* will be significantly lower than *arousal* dimension.

As the labels for these dimensions are not available, we use the circumplex model of affect for speech (shown in Figure 2.4(a)) for mapping the corresponding emotion classes to these two dimensions. We map all emotions in the corresponding databases to low or high *arousal* and negative or positive levels of *valence*. This mapping of emotions and the number of samples per class for these acted emotional speech databases are shown

Table 7.6: Mapping of emotional classes for the three acted speech databases on Arousal and Valence dimension.

	Arousal			
Database	**Low**	#	**High**	#
DES	Neutral, Sad	156	Angry, Happy, Surprised	104
Berlin	Boredom, Disgust, Neutral, Sad	267	Angry, Fear, Happy	268
Serbian	Neutral, Sad	1674	Angry, Fear, Happy	1116
	Valence			
	Negative	#	**Positive**	#
DES	Angry, Sad	156	Happy, Neutral, Surprised	104
Berlin	Angry, Boredom, Disgust, Fear, Sad	150	Happy, Neutral	385
Serbian	Angry, Fear, Sad	1116	Happy, Neutral	1674

Table 7.7: Mapping of five emotional classes for Aibo database onto two cover classes.

Emotion Classes		**Aibo-Mont**	**Aibo-Ohm**
5 Classs	**2 Classes**		
Anger Emphatic	**NEG**ative	2465	3358
Neutral Positive Rest	**POS**itive	5792	6601

in Table 7.6. For the spontaneous speech databases, all of the five classes are common and they have already been mapped onto two binary classes: NEGative and POSitive, for the Interspeech 2009 Emotion Challenge. This mapping of five emotion classes to two binary classes for both parts of Aibo database is given in Table 7.7.

7.7.2 Testing for Covariate Shift

To verify our approach of applying K-S test to check of covariate shift, we first apply this test, feature by feature, in SD–CV setting. As the main assumption of SD–CV is that same data is present in each fold, most of the features should pass the test. We applied K-S test on DES database and out of 3000 features, on average only 3.8% failed to pass. When we applied the same in SI–CV setting, where the test data comes from an unknown speaker, 43.0% features failed the test. We got similar results for other databases.

Table 7.8: Percentage of Uni-Large features failing the Kullback-Liebler test in SD–CV, SI–CVand inter–databasescenarios.

Database	SD–CV	SI–CV	Inter-database
DES	4.8	35.1	95.7
Berlin	6.7	37.6	97.8
Serbian	4.5	77.3	98.0
Aibo-Mont	3.2	82.7	92.2
Aibo-Ohm	3.0	83.6	92.2

These tests verify our assumption that there is a shift in the data when we use an unknown speaker for testing a classifier. This shift can be minimised by using IW-algorithms to improve the generalisation capabilities of a SER system. The increase in the classification accuracy should be more pronounced in the case of SI–CV, where there are very large number of features that failed the K-S test, relative to SD–CV.

To test for the presence/absence of covariate shift in inter–database emotion classification, we have chosen two parts of the spontaneous speech database (Aibo). Although these two parts (Aibo-Mont and Aibo-Ohm) have been recorded in the same wizard-of-oz scenario using the same equipment, we believe that they have covariate shift in them as they were recorded at two different locations with different speakers and acoustic environments. We applied K-S test on the Uni-Large feature set and out of 5311 features tested, 4896 (over 92%) failed the test. This validated our assumption that data for these two classes has covariate shift in it. For inter–database tests on acted databases which are in different languages, an average more than 96% features failed the K-S test. Table 7.8 gives the details of the features failing the K-S test in different settings for all of the databases.

7.7.3 Results on Three Acted Databases

The baseline results on these databases (DES, Berlin, Serbian) using several SVM classifiers have already been established in Chapter 6. In those results, our proposed 3DEC hierarchical classifier performed the best. In this chapter, we have used the same methodology to generate the 3DEC hierarchical classifier from the training data and refer to it as 'standard SVMs' with $c = 0.1$. Note that for 2 class problems or for 4 common classes between the database, this method will not make much difference.

The results of applying SD–CV and SI–CV intra-database classification on the three acted emotional speech databases by compensating for speaker and acoustic differences using the traditional CMN+MLLR method and three methods from transfer learning using all of the classes present in the database are given in Table 7.9. The table shows the UA average accuracies for SD–CV and SI–CV. For DES and Berlin databases, the results using CMN+MLLR or IW-algorithms are better than using standard SVM

Table 7.9: SD–CV and SI–CV percentage UA accuracy on three acted databases using tradition CMN+MLLR and the three IW-algorithms from transfer learning. The numbers in the brackets are the standard deviations using SVM classifier with $C = 0.1$.

	SD–CV			**SI–CV**		
Method	**DES**	**Berlin**	**Serbian**	**DES**	**Berlin**	**Serbian**
SVM	74.6 (3.2)	92.1 (3.1)	**94.6** (0.8)	53.9 (8.2)	79.5 (6.0)	80.1 (4.1)
CMN+MLLR	74.6 (6.2)	87.9 (2.8)	90.6 (2.1)	55.0 (6.2)	78.9 (6.5)	79.9 (2.2)
KMM	**76.5** (5.6)	88.6 (4.0)	92.1 (2.2)	56.1 (7.8)	82.6 (5.9)	**81.3** (2.1)
KLIEP	75.8 (8.8)	86.1 (3.4)	90.8 (3.6)	55.1 (6.3)	78.4 (5.8)	80.1 (3.5)
uLSIF	76.2 (7.5)	**92.3** (2.3)	**94.6** (0.7)	**56.9** (6.4)	**84.9** (5.5)	80.5 (1.6)

classifier. The improvement is much greater in the case of SI–CV as compared to SD–CV due to the reasons already discussed.

The results of applying SD–CV and SI–CV intra-database classification for *arousal*, *valence* and 4 common classes among all of the acted databases are given in Table 7.10(a) and Table 7.10(b) respectively. On average, we get 97.8% and 84.0% SD–CV UA accuracy for *arousal* and *valence* dimensions respectively. From these results it is clear that the *arousal* dimension is much easier to recognise as compared to the *valence* dimension. Worst results for *valence* recognition are obtained for the DES database (74.5% UA) while for the other two, they are above 90% UA which is very good. This means that for this database, it is not only difficult to separate *angry* from *happy* which have positive *valence*, but these two are also not very easily separable from *neutral* and *sad* emotions in the *valence* dimension. For four common classes among the three database, best accuracy is obtained for the Serbian database (91.6% UA) while worst results are obtained for the DES database (76.0% UA).

Interestingly, the classification results for SI–CV are very close to SD–CV especially by using CMN+MLLR and IW-algorithms. In some of the cases (DES and Berlin) by using these algorithms we get very large improvements in comparison to using the standard SVM classifier. This actually fits with the theoretical basis of these methods as there is a larger room for improvement for SI–CV than for SD–CV, which is seen from the results.

An important observation is that the average results for *arousal* and *valence* recognition by CMN+MLLR and IW-algorithms for all of the databases are better than the results of standard linear SVM. This means that by using methods that explicitly compensate for the speaker and environmental differences, improves the results significantly.

The CMN+MLLR algorithm does improve the classification results in comparison to the standard SVM. However, when compared against the three IW-algorithms, it only performs better in 1 out 18 SI–CV and SD–CV experiments. Generally, we get better results by applying IW-algorithms which compensate for the covariate shift in the data.

Table 7.10: SD–CV and SI–CV intra-database percentage UA accuracy on three acted databases for *arousal*, *valence* and 4-common classes using traditional CMN+MLLR method and the three IW-algorithms from transfer learning. The numbers in the brackets are the standard deviations using SVM classifier with $C = 0.1$.

(a) SD–CV Intra-database classification results.

	DES			**Berlin**			**Serbian**		
Method	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**
SVM	95.5 (3.2)	71.8 (7.8)	74.6 (8.7)	95.9 (2.7)	93.6 (2.8)	84.8 (4.1)	99.5 (0.3)	91.2 (0.9)	91.3 (2.0)
CMN+MLLR	97.0 (3.0)	70.5 (6.4)	74.7 (8.5)	96.6 (4.6)	92.9 (2.7)	84.9 (4.1)	**99.5** (0.1)	91.0 (1.1)	91.1 (2.6)
KMM	**99.2** (1.9)	75.2 (6.6)	76.9 (10.0)	97.2 (3.3)	93.3 (3.1)	85.2 (5.0)	98.1 (0.7)	**94.4** (0.8)	91.8 (2.22)
KLIEP	97.4 (4.7)	74.1 (8.4)	76.4 (9.7)	97.2 (2.7)	**93.9** (2.0)	**87.6** (2.6)	99.5 (0.4)	91.4 (0.5)	91.6 (3.6)
uLSIF	97.2 (3.0)	**75.8** (8.5)	**77.2** (7.1)	**97.4** (3.4)	92.1 (1.5)	86.0 (4.1)	99.5 (0.4)	91.1 (1.1)	**92.1** (1.9)
Mean	**97.3**	**74.5**	**76.0**	**96.8**	**93.2**	**85.7**	**99.2**	**91.8**	**91.6**

(b) SI–CV Intra-database classification results.

	DES			**Berlin**			**Serbian**		
Method	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**
SVM	88.6 (3.9)	76.5 (7.6)	76.0 (8.6)	93.3 (6.1)	92.1 (1.5)	84.8 (2.7)	96.4 (3.9)	88.5 (2.6)	81.2 (7.5)
CMN+MLLR	89.0 (4.5)	77.1 (9.3)	76.0 (7.5)	94.8 (3.0)	92.3 (3.1)	87.5 (4.1)	96.2 (2.4)	90.5 (3.2)	83.5 (6.9)
KMM	**91.5** (8.3)	**83.1** (9.0)	77.9 (11.5)	98.3 (1.5)	93.3 (3.0)	91.6 (1.7)	**97.6** (2.8)	91.6 (3.8)	84.1 (6.6)
KLIEP	91.3 (6.5)	82.6 (8.5)	76.4 (3.7)	97.9 (1.5)	**93.9** (2.4)	**92.3** (1.7)	96.9 (3.5)	90.6 (2.3)	**84.7** (6.1)
uLSIF	90.0 (6.7)	81.1 (6.9)	**78.8** (7.1)	**98.3** (1.2)	93.6 (2.8)	89.1 (1.8)	97.0 (1.9)	**91.7** (5.3)	84.1 (6.5)
Mean	**90.1**	**80.1**	**77.0**	**96.5**	**93.0**	**89.1**	**96.8**	**90.6**	**83.5**

Out of the three IW-algorithms, uLSIF performs best in 7 out of 18 experiments. This shows that just like CMN+MLLR, IW-algorithms can also be successfully used to compensate for the mismatch between the training and testing data caused by different speakers.

The results of inter–database emotion classification are given in Table 7.11. They are obtained by applying leave-one-database-out cross validation. The database marked at the top of each column was used for testing while the remaining two were used for training the classifiers. It can be observed that inter–database accuracy for *arousal*, *valence* and four common classes is lower than intra–database classification accuracy. This is very much expected as the recording environments and speakers for the training and testing data are separate and different from each other. This kind of situation is the one which will be faced by any practical SER system. In such a situation, one has to apply some methods to compensate for the mismatch. From the results shown in Table 7.11, one can see that CMN+MLLR does significantly increase the classification accuracy as compared to standard SVM. However, the increase in classification accuracy is less as compared to the IW-algorithms. Out of the three IW-algorithms, uLSIF based classification performs best in 7 out of 9 experiments. Similar results were seen on the 2D toy data shown in Figure 7.4 in which uLSIF is performing the best for small values of penalty factor C. Hence, we declare uLSIF as the best out of the three tested IW-algorithms.

These results are very interesting as all of the three databases tested are in different languages. Although German and Danish languages belong to the same family of Germanic languages, they share some similarities. The Serbian is a Slavonic language which does not belong to the same family. On average, we get 78.3% and 57.3% UA accuracies for *arousal* and *valence* recognition by testing on the database which has different speakers, recording environments and different language than those used for training the classifiers. These are very good results considering such large differences between the training and testing datasets. Especially, the UA for inter–database *arousal* recognition is very high and UA accuracy for inter–database *valence* recognition is also above chance level. Best inter–database classification results are obtained for testing on the Serbian database. As mentioned earlier, this database does not belong to the family of Germanic languages so the expected results should have been opposite. However, average accuracy on this database is generally very high which is the reason for these results. Secondly, all of these databases contain European languages. So there are some cultural aspects common between them. These arguments can explain these results.

These experiments show that there are some aspects of emotions which are *universal across several languages*. Even if the classifier does not have any information about the test language, it can still get quite reasonable results, better than random guessing. These results also validate our assumption that by using different databases for training and testing, which have different speakers, acoustic environments and languages as well,

Table 7.11: Inter–database percentage UA accuracy on three acted databases for *arousal*, *valence* and 4-common classes using traditional CMN+MLLR method and the three IW-algorithms from transfer learning.

Testing on →	**DES**			**Berlin**			**Serbian**		
Method	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**	**Arousal**	**Valence**	**4-Class**
SVM	71.4	50.8	40.5	72.9	49.2	39.5	82.7	64.2	63.3
CMN+MLLR	74.1	50.4	41.3	74.6	50.0	40.0	83.9	67.4	65.0
KMM	75.0	51.5	42.8	75.0	50.1	43.6	84.9	66.7	**65.5**
KLIEP	75.4	51.5	43.3	73.3	58.2	41.3	85.8	**69.3**	65.2
uLSIF	**82.4**	**51.7**	**44.7**	**75.8**	**58.4**	**46.1**	**87.8**	69.1	64.5
Mean	**75.7**	**51.2**	**42.5**	**74.3**	**53.2**	**42.1**	**85.0**	**67.3**	**64.7**

introduces a shift in the data which can be compensated by traditional methods used in ASR systems as well as IW-algorithms. Generally, IW-algorithms perform better than CMN+MLLR, and out of the three algorithms tested, uLSIF performs the best.

7.7.4 Results on Spontaneous Emotional Speech Database

Results of SD–CV and SI–CV classification using CMN+MLLR and IW-algorithms on the Aibo-Mont and Aibo-Ohm are given in Table 7.12. From the results one can see the significant gain achieved by using methods that can compensate for speaker differences. These gains are not very large for SD–CV as there is a chance that data from the speaker who is being tested may already be present in the training dataset. However, the gain of using CMN+MLLR and IW-algorithms is significant in the case of SI–CV for which the validation setup guarantees that no information about the current speaker is available to the classifier. In such a case, standard SVM classification does not perform as well as the other algorithms which compensate for the different speakers. For these two emotional speech databases containing spontaneous data, CMN+MLLR method significantly improves the UA classification accuracy. However, IW-algorithms generally perform better than CMN+MLLR. Out of the three IW- algorithms tested, uLSIF gives best accuracy for 6 out of 8 tests.

Both of these databases contain spontaneous speech of children interacting with a robot in German language. However, the two databases were recorded at two different schools. Although the same equipment was used for the recording sessions, the room acoustics can not be guaranteed. Therefore we believe that this will cause a covariate shift in the data.

The results of inter–database emotion classification are given in Table 7.13 where the database marked at the top of each column was used for testing while the other for training the classifier. We can see a significant improvement in the UA classification accuracy by explicitly compensating for the speaker and acoustic differences between the training and testing datasets. The uLSIF base IW-algorithm performs best in 2 out of the 4 experiments. Therefore in this case we can not declare a clear winner. Generally, IW-algorithms perform better than the standard CMN+MLLR algorithms. Interestingly, these results obtained by using uLSIF importance weighting algorithm are even better than the one reported in Lee *et al.* (2011). This is the paper that was declared as winner of the Interspeech 2009 Emotion Challenge. They have reported 41.6% UA average accuracy for five classes using hierarchical structure of binary decision tree while training on Aibo-Ohm and testing on Aibo-Mont.

The results on the two spontaneous emotional speech database prove our argument that although these two databases have been recorded under similar conditions, they are not exactly the same. This is clear by the increased classification accuracy obtained by

applying algorithms that compensate for the acoustic and speaker differences. As we have seen that IW-algorithms clearly out performs traditional CMN+MLLR algorithm, it supports our claim that these algorithms can be used to compensate for the differences induced by different speakers and acoustic environments.

7.8 Summary

In this chapter, we have tackled a very difficult but important topic of improving the generalisation capabilities of SER for intra and inter-database emotion classification; a topic which has been generally ignored by the emotion recognition community. In this chapter we have tested two traditional methods (CMN and MLLR) that are often used in ASR systems to compensate for speaker and environment differences. We have shown that these differences can be modelled as a covariate shift. We have proposed to use a few importance weighting algorithms to compensate for this covariate shift. Our experiments show that these algorithms can successfully reduce speaker and acoustic differences for intra and inter–database emotion classification. They perform better than traditional CMN+MLLR methods for both intra– as well as inter–database emotion classification. Very interesting results are obtained for inter–database classification when we test on the databases that are in different languages than the training ones pointing towards the em universal nature of emotions. The experimental results in this chapter show that importance weighting algorithms that compensate for the covariate shift can be successfully used for improving the generalisation capabilities of speech emotion recognition systems.

when we test on databases which are in different languages than the training ones. The experimental results show the *universal* nature of emotions and importance weighting algorithms that compensate for the covariate shift can be successfully used for improving the generalisation capabilities of speech emotion recognition systems.

Table 7.12: SD–CV and SI–CV percentage UA accuracy on the spontaneous database using tradition CMN+MLLR and the three IW-algorithms from transfer learning. The numbers in the brackets are the standard deviations.

	SD–CV				**SI–CV**			
	Aibo-Mont		**Aibo-Ohm**		**Aibo-Mont**		**Aibo-Ohm**	
Method	**2-Class**	**5-Class**	**2-Class**	**5-Class**	**2-Class**	**5-Class**	**2-Class**	**5-Class**
SVM	68.0 (1.2)	49.4 (3.3)	70.6 (2.2)	48.1 (1.4)	65.9 (6.4)	43.4 (6.0)	67.5 (6.2)	44.8 (2.1)
CMN+MLLR	68.2 (0.9)	51.3 (3.2)	70.6 (2.3)	48.2 (1.9)	66.4 (6.2)	44.0 (5.2)	69.0 (5.7)	46.2 (2.5)
KMM	69.2 (0.7)	52.7 (3.0)	72.7 (4.8)	**50.0** (1.2)	65.8 (5.7)	45.1 (4.8)	68.2 (5.4)	48.0 (2.1)
KLIEP	70.1 (0.8)	52.9 (3.1)	72.1 (3.2)	48.8 (1.5)	66.1 (6.1)	**46.3** (4.2)	70.4 (4.7)	46.8 (2.7)
uLSIF	**72.6** (0.6)	**53.9** (2.9)	**73.5** (1.8)	49.4 (1.6)	**67.5** (5.2)	46.2 (3.9)	**71.0** (4.6)	**49.4** (2.0)
Mean	**69.6**	**52.0**	**71.9**	**48.9**	**66.3**	**45.0**	**69.2**	**47.0**

Table 7.13: Inter–database percentage UA accuracy on the spontaneous speech database using traditional CMN+MLLR method and the three IW-algorithms from transfer learning.

Testing on →	**Aibo-Mont**		**Aibo-Ohm**	
Method	**2-Class**	**5-Class**	**2-Class**	**5-Class**
SVM	67.3	38.0	67.0	36.4
CMN+MLLR	70.7	39.4	70.0	40.5
KMM	**72.8**	42.2	71.2	40.6
KLIEP	72.7	40.5	**72.4**	41.3
uLSIF	72.5	**42.7**	71.6	**41.8**
Mean	**71.2**	**40.6**	**70.4**	**40.1**

Chapter 8

Conclusions and Future Work

In this thesis, we have looked at different acoustic feature sets that can be used for emotion recognition from speech. We have given the details of an extensive feature set that was generated using brute force. Out of this large feature set, we found energy and spectrum based features to be performing the best. After testing several wrapper and filter based feature selection/ranking methods, we have proposed a novel feature ranking scheme based upon preferential Borda ranking. This method can strike a balance between accurate but computationally intensive wrapper methods and less accurate but computationally less intensive filter methods. In our classification tests, features selected by hard decision based Borda ranking outperformed the rest. Using the hard decision Borda ranking, we have searched for a universal feature set that performs reasonably well on all/any emotional speech database. These universal features proved their effectiveness in the inter–database emotion recognition tests.

We have used a data driven approach to map the emotions onto a two dimensional space using non-metric multi-dimensional scaling. This representation can be interpreted in terms of the valence-arousal model of emotions. From our tests on acted as well as spontaneous emotional speech data, we can identify *arousal* clearly in the transformed data while *valence* is not well represented. From these results we can draw the following conclusions: either the valence-arousal model has a basic flaw and *valence* is not present in the real world data or the *valence* and *arousal* dimensions are not orthogonal to each other.

In our experiments on recognising these two dimensions from several emotional speech databases, we found the classification accuracy on *valence* to be much lower than *arousal*, as expected. However, this accuracy is much higher than the chance level, strongly indicating the presence of *valence* in emotional speech. We conclude that this dimension is present in emotional speech data however, it is not orthogonal to *arousal* as is usually assumed.

Using NMDS plots, we have proposed a hierarchical classifier (3DEC), for which the hierarchical structure is determined by a data driven approach. This 3DEC hierarchical classifier gives significantly better results than traditional and state-of-the-art hierarchical classifiers.

In the last part of this thesis, we have looked at methods to improve the generalisation capabilities of a SER over varied speakers and acoustic environments. This is a very difficult but important topic in emotion recognition, which has been mostly over looked. To address this issue, first we have tried some traditional methods like CMN and MLLR, used in ASR systems. These methods do improve the average accuracy. However, in this thesis, we have identified the differences due to speakers and acoustic environments as a covariate shift between the training and testing data. To compensate for this shift we have used three methods from the emerging field of transfer learning. All of these methods calculate the importance weights to shift the classifier towards that training data which gives better representation of testing data. This is the first time such algorithms have been used in speech or emotion recognition. Using these importance weighting algorithms, we were able to improve the classification accuracy on all of the databases tested. Our results show that IW-algorithms can be used to match the differences between the training and testing data due to different speakers and acoustic environments.

To test the effectiveness of IW-algorithms, we have applied them on inter–database emotion classification. This setting presents even harder challenges for the SER system as it has to deal with the speakers and environments that have not been seen before. This area of research is very new in the context of emotion recognition. We have used IW-algorithms to compensate for the speaker and acoustic environment differences and the results show that these methods outperform the standard CMN and MLLR. These IW-algorithms have proved their effectiveness in cross-lingual tests. These tests also prove the universal nature of emotions across different languages.

Based upon the results presented in this thesis, we can conclude the following:

- The proposed Borda feature ranking method is effective in choosing overall best ranked features using several feature ranking methods.
- The proposed data driven method to guide the hierarchical structure for multi-class emotion recognition classifier is also very effective and improves the results significantly.
- Importance-weighting algorithms can be used to compensate for the covariate shift in the data caused by different speakers and acoustic environments.
- There are aspects of emotions which are *universal* across different languages.

Based upon our work in this thesis, there are many avenues that can be explored further.

Multi-modal Online Emotion Recognition

In this thesis, we have only used acoustic features for emotion recognition as we had databases in which the speech is already separated into sentences. However, for a dialogue system, semantic and discourse information can be really helpful. These source of information need to be considered for developing complete SER systems for day to day dialogue systems.

Another possible extension of this work is in the field of multi-modal emotion recognition. SER algorithms proposed in this thesis can be combined with facial expression recognition techniques to enable machines to recognise emotions more accurately. However, this task has to be done carefully as in human beings, facial expressions dominate verbal expressions during communication. For this task further psychological studies are to be done to understand the ways and methods that humans use for achieving communication of emotions.

In our work we have focused on offline SER systems. The ultimate goal is to develop online SER systems. In such a system all the processing has to be done in real time. This is definitely not an easy task and there are many things that need to be considered. As we have shown machine recognition is coming very close to human accuracy for acted speech but this is not true for the real life (simultaneous) speech. The accuracy on the real life speech has to be improved.

For real life emotions, one would like to calculate as few features as possible. We have identified some of the important features for an SER system for real life speech but further studies are to be done for the search of minimal features to make online emotion recognition a practical possibility.

Multilingual Universal Feature Set

In our work, we have tried to discover the universal features that are helpful for recognising emotions across several languages. Our work is one attempt towards achieving this goal; however this area of research is very open. It has to be explored thoroughly. Only then one can claim to have developed language independent emotion recognisers.

In regard to developing language independent features for emotion recognition, psychologists can play a very important and useful role. With the help of psychology, we can understand how humans perform this task of detecting emotions even without the knowledge of language e.g., infants can recognise emotions even before they can speak. Such information is going to be vital for complete and thorough understanding of human emotions which can then be translated for machine use.

Globally Optimum Feature Set

All wrapper or filter feature selection methods try to find a set of features that perform better than others under certain conditions but they can not guarantee that the selected feature set is the globally optimum solution. Searching for a

globally optimum set of features has always been a computationally un-feasible task. The task is to select one combination out of 2^D possible combinations of features. To solve this problem, Nguyen *et al.* (2009) proposed to model any feature selection algorithm as a polynomial mixed 0-1 fractional programming (PM01FP) problem, where 0-1 corresponds to the absence or presence of the corresponding features in the selected feature set. Chang (2001) has proposed a technique to linearise polynomial fractions which can be used to transform the PM01FP to a mixed 0-1 linear problem. This mixed 0-1 linear problem can be easily solved by any linear optimiser to get a global solution. Nguyen *et al.* (2009) have applied this method on mRMR and correlation based feature selection methods and shown improved performance on several publicly available datasets.

Potentially this means that we can search for a globally optimum feature set at a running cost of a linear optimisation technique. This is a huge improvement in computation costs and the solution is also globally optimum. These techniques need to be further investigated and tested on several available datasets to verify their effectiveness.

Application of IW-Algorithms in Other Domains

Compensating covariate shift using IW-algorithms has a very useful application in the fields like brain computer interfacing and intrusion detection systems. Our initial experiments on BCI Competition III held in 2005 (not presented in this thesis) using IW-algorithms proved very successful. Similarly, covariate shift modelling methods have been successfully applied by Farran *et al.* (2010) on intrusion detection systems. Our tests of applying IW-algorithms to reduce the speaker and environment differences for emotion recognition have been very successful. It would be very interesting to see their effect on ASR and speaker recognition systems. These methods need to be applied on further experimental datasets to verify their strengths.

Improved IW-Algorithms

All of the three importance weighting algorithms discussed in this thesis try to minimise the distance between the distributions. However, these methods do not guarantee to preserve data variance properties. In this regard, kernel PCA does guarantee to preserve the maximum variance of the data. There is a requirement of developing transfer learning algorithms that minimise the distance between the distributions while preserving the data variance. Pan *et al.* (2011) is one effort in this direction. However, this area of research is very new and can benefit from further research.

Online IW-Algorithms

The IW-algorithms have been developed for problems where a large amount of unlabelled testing data is available and importance weights can be calculated using

limited training data. However, these algorithms have been developed for offline processes. This means that they can not be applied to runtime problems where processing time is of essence. If we wish to make these algorithms integral to SER systems, they have to be modified to make them work online. This is where IW-algorithms are lacking in comparison to CMN and MLLR algorithms.

If we consider the example of kernel mean matching, it is a quadratic optimisation problem. We know that SVMs are also solved as a quadratic optimisation problem. There are several methods like Vijayakumar and Wu (1999) and Platt (1998) which have been developed to sequentially calculate SVM parameters for online applications. These methods can be used to adapt importance weighting algorithms and make them suitable for online applications.

In this thesis, we have presented our efforts towards developing automated emotion classification systems. However, there is much more work that needs to be done before this technology can become part of our daily life.

Appendix A

Pitch Extraction Methods

Several methods for pitch extraction have been developed over the time. In this section we shall explain some of the algorithms that were experimented with for pitch extraction.

A.1 Autocorrelation Function

Correlation of two signals is the similarity measure of one signal with the other whereas autocorrelation of a signal is the similarity measure with itself. This is the simplest algorithm used for pitch detection which gives reasonably acceptable results. It exploits the fact that a periodic signal will repeat itself after some definite interval which is the fundamental period (T_0) of the signal. This assumption will also hold when we are dealing with speech signals which are pseudo-periodic signals. The reciprocal of this fundamental period will give the pitch of the signal. Mathematically, the autocorrelation function (ACF) of a signal x of infinite extent is defined by:

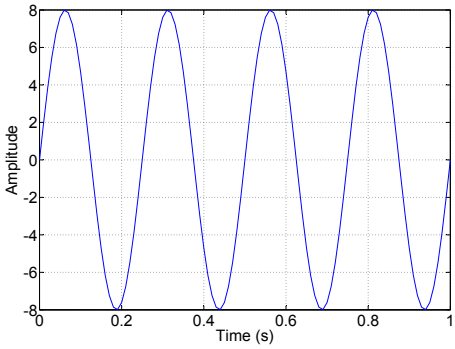
$$r(\tau) = \sum_{j=-\infty}^{\infty} x_j x_{j+\tau} \quad \tau = -\infty \ldots -1, 0, 1, \ldots, \infty \tag{A.1}$$

where $r(\tau)$ is the autocorrelation function of lag τ.

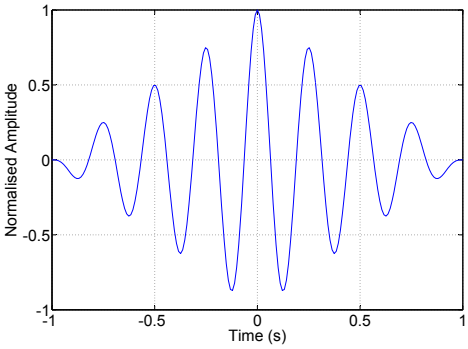
To calculate the continuously changing pitch period of the signal, the input speech signal is divided into small frames of 15 ms to 25 ms, i.e., for $f_s = 10$ kHz each frame will consist of 150 to 250 samples. Each frame is processed independently and the results give the pitch contour of the speech signal. For a framed signal of length W, the autocorrelation equation can be written as:

$$r_t(\tau) = \sum_{j=t}^{t+W-1} x_j x_{j+\tau} \quad \tau = \tau_{min}, \ldots, -1, 0, 1, \ldots, \tau_{max} \tag{A.2}$$

where $r_t(\tau)$ is the ACF for lag τ for frame number t.



(a) Sine signal of frequency 4 Hz.



(b) Normalised autocorrelation function of sine wave.

Figure A.1: Auto correlation function of a sine signal (a) a sine signal of 4 Hz; (b) normalised autocorrelation function showing peaks after every T_0.

If the signal is periodic with period T_0, then:

$$r(\tau) = r(\tau + T_0) \tag{A.3}$$

Figure A.1(a) shows a sine signal of 4 Hz and its ACF is shown in Figure A.1(b). The ACF has been normalised so that the highest peak has the maximum amplitude of 1. As can be seen from the figure, the ACF is an even function of time, therefore the positive half of the ACF should be sufficient for any algorithm to deduce results. We shall consider the second half of the even ACF for the rest of this report. Peaks in the ACF show the locations in time when a signal repeats itself. In other words, after every T_0 seconds, we should find a peak in the ACF. The distance between the starting point and the first largest peak is T_0 for that signal and $1/T_0$ will give the pitch of the speech signal.

A.2 Average magnitude difference function (AMDF)

This method of average magnitude difference function (AMDF) for pitch extraction was introduced by Shaffer *et al.* (1973) and is explained by Ross *et al.* (1974). The later authors proposed that a variation of autocorrelation function, i.e., difference function, which can be used for determining periodicity of a signal. The AMDF of a framed signal of length W is mathematically defined as:

$$d_t(\tau) = \frac{1}{W} \sum_{j=t}^{t+W-1} |x_j - x_{j+\tau}|, \quad \tau = 0, 1, \ldots, \tau_{max} \tag{A.4}$$

where vertical bars denote the magnitude of the difference $x_j - x_{j+\tau}$, i.e., difference of original signal and its shifted value. Thus an averaged difference signal $d_t(\tau)$ is formed

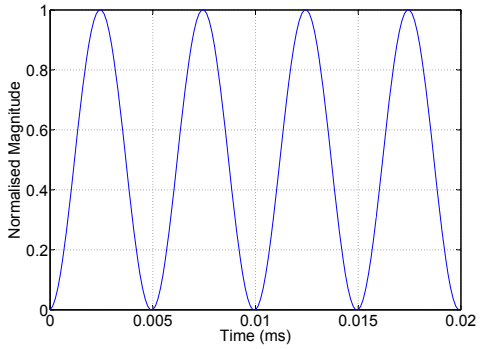


Figure A.2: Difference function of a sine wave of 200 Hz.

by delaying the jth input speech frame by τ samples to form a delayed version $x_{j+\tau}$ of x_j, and summing the magnitude of the difference between sample values.

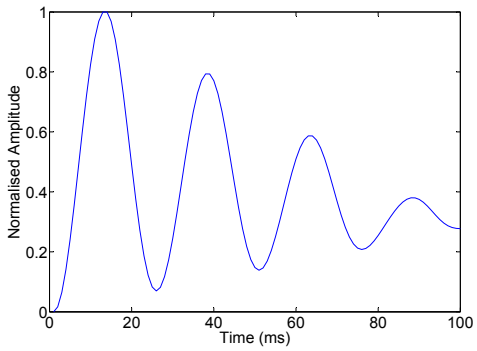
AMDF is related to the autocorrelation method in the sense that both compare a signal to itself shifted. In the AMDF method, the comparison is done using differences rather than products. Figure A.2 shows the results of applying eqn. (A.4) on a sine signal of 200 Hz. It is evident that the difference signal will always be zero at $\tau = \phi$, and is observed to have deep nulls at delays corresponding to the pitch of the signal. Therefore, instead of looking for maxima in the difference function, we shall be looking for minima which will give us the pitch period.

Based upon this method, we find a number of publications with small variations, which primarily use the difference function for extracting pitch. The YIN algorithm (Cheveigne and Kawahara, 2002) and Tartini project (McLeod and Wyvill, 2005) have used the same difference function with small variations in their definitions.

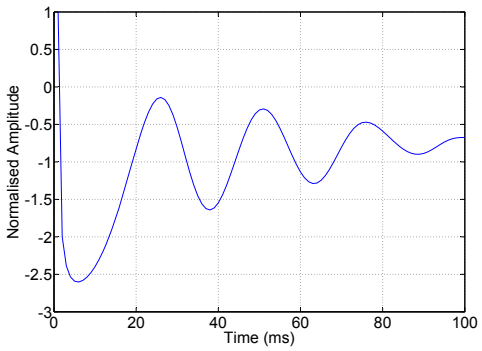
A.3 YIN Algorithm

As was mentioned previously, the YIN algorithm (Cheveigne and Kawahara, 2002) is based upon the difference function for calculating the periodicity of a signal. Instead of using AMDF, YIN uses the square difference function (SDF) for calculating periodicity in the signal which is mathematically defined as:

$$\begin{aligned} d'_t(\tau) &= \sum_{j=t}^{t+W-1} (x_j - x_{j+\tau})^2 \quad \tau = 0, 1, \ldots, \tau_{max} \\ &= \sum_{j=t}^{t+W-1} \left(x_j^2 + x_{j+\tau}^2 - 2x_j x_{j+\tau}\right) \end{aligned}$$



(a) Square difference function



(b) Cumulative mean normalised difference function

Figure A.3: Results of YIN algorithm applied on sine wave of 400 Hz (a) SDF as proposed in YIN algorithm; (b) cumulative mean normalised difference function.

Using eqn. (A.2), we get:

$$d'_t(\tau) = r_t(0) + r_{t+\tau}(0) - 2r_t(\tau) \tag{A.5}$$

where the first two terms are the energy terms and the last term is the autocorrelation of signal. Therefore, the SDF function automatically embeds the autocorrelation function in itself.

SDF is also zero at $\tau = \phi$, and is observed to have deep nulls at delays corresponding to the pitch period of the signal. To normalise the SDF, the YIN algorithm proposes to use cumulative mean normalised difference function given by:

$$n_t(\tau) = \begin{cases} 1 & \text{if } \tau = 0 \\ n_t(\tau) / \left[(1/\tau) \sum_{j=t}^{t+W-1} d'_t(j) \right] & \text{otherwise} \end{cases} \tag{A.6}$$

This function, $n_t(\tau)$, differs from $d'(\tau)$ in that it starts at 1 rather than 0 and tends to show peaks corresponding to the pitch period of the signal. Figure A.3(a) and Figure A.3(b) show results of SDF and normalised SDF, respectively, as proposed by Cheveigne and Kawahara.

A tool named 'Tartini' has been developed at the University of Otago, New Zealand for practical music analysis for singers and instrumentalists. This tool is available online for free at `http://miracle.otago.ac.nz/postgrads/tartini/download.html`. It gives the real-time feedback of pitch contours for visualising intonation, vibrato shape, loudness graphs, tuning or just which note is being played, to help analyse dynamics and harmonic structure of a note describing timbre. The algorithm for pitch extraction is explained in McLeod and Wyvill (2005). It uses a slight variation in YIN algorithm

for normalising the SDF by rewriting eqn. (A.5) as:

$$d'_t(\tau) = m_t(\tau) - 2r_t(\tau) \tag{A.7}$$

The normalised SDF can be written as:

$$\begin{aligned} n'_t(\tau) &= 1 - \frac{m_t(\tau) - 2r_t(\tau)}{m_t(\tau)} \\ &= \frac{2r_t(\tau)}{m_t(\tau)} \end{aligned} \tag{A.8}$$

and normalised SDF $n'_t(\tau)$ ranges from $+1$ to -1, where $+1$ means perfect correlation, 0 means no correlation and -1 means perfect negative correlation.

A.4 Sub-harmonic Summation

This idea of pitch detection by sub-harmonic summation is based upon the principle that a speech sound is formed by the combination of a fundamental frequency and its shifted and weighted harmonics. Therefore, if we transform a signal into Fourier domain, and compress the signal with different fractions, i.e., down sample the signal, product of compressed signals with the original will give us the fundamental frequency.

On a linear frequency scale, we have to take the product of down-sampled versions of the original signal. In Hermes (1988), the author explains that if we take log of linear frequency domain, then down-sampling in the frequency domain will change to a simple shift in the logarithmic domain and multiplication will change to simple addition as given by the following equations:

$$H(f) = \prod_{n=1}^{N} h_n P(nf) \tag{A.9}$$

and on logarithmic scale with $s = \log_2 f$

$$H(s) = \sum_{n=1}^{N} h_n P(s + \log_2 n) \tag{A.10}$$

where n is the compression factor and h_n is the scaling factor equal to 0.84^{n-1}. As can be seen from the equation, scaling and product of sub-harmonics in frequency domain change to simple shifting and summation in the logarithmic domain. A good feature of this method is that it is computationally inexpensive but on the other hand the results are dependent upon the number of subharmonics used for calculation.

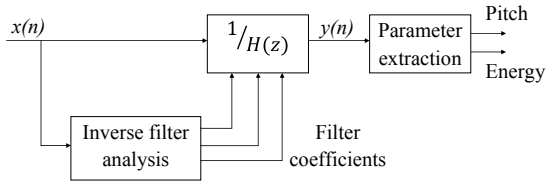
A.5 Linear Predictive Analysis

Linear predictive coding (LPC) is a digital method for encoding an analogue signal at very low rate with an acceptable voice quality. LPC was introduced in the early 1970's and the first paper specifically on LPC was Atal and Hanauer (1971). A complete descriptive tutorial of LPC can be found in Makhoul (1975). LPC achieves high compression rates by sending only the speech parameters to the receiver instead of the whole speech signal. In 1984, it was officially published as an encoding standard by the United States Department of Defence in Federal Standard 1015.

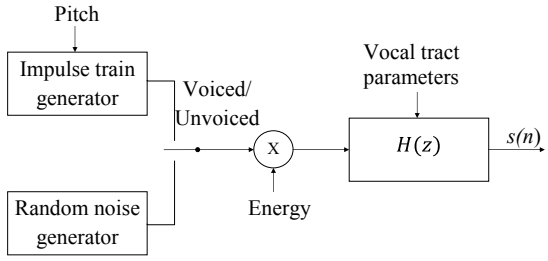
LPC is a model-based speech coder that mathematically approximates the human speech production mechanism to extract speech parameters. It approximates the current speech sample based upon the linear combination of previous samples. As LPC is a model-based approach, it will never be able to reproduce the original speech signal exactly.

LPC uses the source-filter speech production model which has been explained earlier. It assumes that the speech signal is produced by pulses of waves produced by a random source generator at the end of a tube. The random source generator are the vocal folds and the pulses of waves are characterised by their loudness and frequency (pitch). The vocal tract forms the tube, which is characterised by its resonance frequencies called formants. This tube is modeled as an all-pole filter and filter coefficients are calculated by solving system of linear equations and are used to build an all-pole filter $H(z)$ that estimates the response of vocal tract. If $x(n)$ is the input signal, and $1/H(z)$ is the inverse filter, then $y(n)$ is the residual signal from which the effect of the vocal tract has been removed. This signal only contains the information about the source of $x(n)$ and is used to extract pitch and energy information of the source. This whole process of extracting speech parameters from sound input using LPC is called LPA and is shown in Figure A.4(a). Autocorrelation is used to extract the pitch information from the residual signal and this method of extracting pitch from residual signal of LPC using autocorrelation is also called simplified inverse filter tracking (SIFT) algorithm explained by Markel (1972).

When LPC is used as a speech coder, only the filter coefficients, pitch and energy information are sent to the receiver where speech is regenerated based upon the received parameters. A block diagram for speech production at the receiver side based upon these parameters is shown in Figure A.4(b). It is quite evident from our discussion that LPA can be used as a pre-processor for any pitch extraction algorithm to remove the effect of the vocal tract.



(a) Block diagram of parameter extraction using LPC.



(b) Block diagram of speech generation at receiver using LPC parameters.

Figure A.4: Block diagram of (a) parameter extraction using LPC; (b) speech generation at receiver.

A.6 Cepstrum Analysis

A fundamental deduction from the source-filter model is that if we consider short segments of the speech signal, then each segment of the speech signal can be modelled as having been generated by passing an excitation signal which is a pseudo-periodic impulse train or a random noise through a filter. Since the excitation signal and impulse response of the filter are combined by convolution, the problem of speech analysis can also be viewed as separating the components of the convolved signal which is also referred as deconvolution. This idea is the basis of cepstrum analysis of a speech signal.

We can find two different definitions of cepstrum analysis, namely, power cepstrum and complex cepstrum which have been quite commonly confused among researchers. The power cepstrum, which is also referred as only 'cepstrum', was initially used in Bell Laboratories by Bogert in early 1960 (Noll, 1967) during his work on seismic signals. He observed "periodic" ripples in the spectra of seismic signals and that these ripples were the characteristic of the spectra of any signal consisting of itself with an echo. The frequency spacing of these ripples equals the reciprocal of the difference in time intervals of the two waves. A spectrum analysis of the log spectrum then could be performed to determine the frequency of the ripples. Bogert, Tukey and Healy published this idea as an internal Bell Laboratories memorandum which was later published in a book by Rosenblatt (1963, Chap. 15).

In 1962, this idea was taken up by researchers in speech processing who were trying to separate the effect of vocal tract from excitation signal to determine the pitch. Here we find two different works done by two different researches which later on gave rise to two

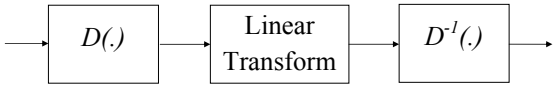


Figure A.5: Block diagram of homomorphic deconvolution.

definitions of cepstrum transform. The definition of cepstrum given by Noll (1967) was "power spectrum of the logarithm of the power spectrum" which can be mathematically written as:

$$C(\tau) = \mathcal{F}\{\log|\mathcal{F}\{x(n)\}|^2\} \tag{A.11}$$

where $\mathcal{F}$ is the Fourier transform of the signal. As this definition utilises the double forward transforms, all the values are real and this is the reason it is called 'power cepstrum'.

The second definition of cepstrum came from the work of Oppenheim and Schafer (1968) on homomorphic deconvolution for signals. To explain homomorphic deconvolution, consider the following example. Let $s(t)$ be a signal consisting of the convolution of two components $s_1(t)$ and $s_2(t)$ such that $s(t) = s_1(t) * s_2(t)$, where '$*$' represents the convolution between the two components. We seek an invertible transform D such that

$$D[s_1(t) * s_2(t)] = D[s_1(t)] + D[s_2(t)] \tag{A.12}$$

which is shown in Figure A.5 where D^{-1} is the inverse transform D. Now, the two convolved components are combined together linearly which can be separated using simple filtering processes.

When this theory is extended to deconvolution of a speech signal which is the combination of excitation signal $p(n)$ and the effect of the vocal tract represented as a filter response $h(n)$, the invertible transform is the combination of Fourier transform and logarithm. The FFT transforms the convolution into multiplication, given as:

$$S(j\omega) = P(j\omega)H(j\omega) \tag{A.13}$$

By using logarithm to convert the multiplied operation into summation, we have:

$$\hat{S}(j\omega) = \log S(j\omega) = \log P(j\omega) + \log H(j\omega) \tag{A.14}$$

Here, we know that Fourier transform of a signal is a complex function. Therefore, we have to define the logarithm of a complex value. If we have a complex number $m = re^{j\omega}$, then

$$\log(m) = \log|m| + j\theta(j\omega) \tag{A.15}$$

Applying this definition of complex logarithm on eqn. A.15, we get:

$$\hat{S}(j\omega) = \log|P(j\omega)| + \log|G(j\omega)| + j\theta_P(j\omega) + j\theta_G(j\omega) \tag{A.16}$$

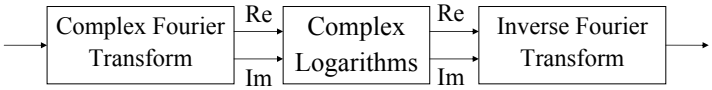
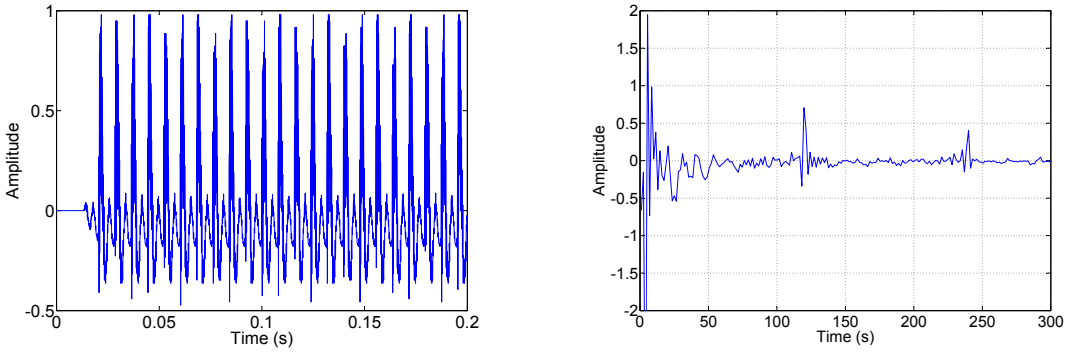


Figure A.6: Block diagram of complex homomorphic deconvolution.



(a) Time-domain speech signal.

(b) Cesptrum transform of the signal

Figure A.7: (a) speech signal in which the speaker has said the vowel '*a*' as in the word 'father'; (b) cepstrum transform of the signal showing the separation of the excitation signal and the vocal tract filter.

where the first two are the magnitude terms and the last two are the complex angle terms. If we take the inverse Fourier transform of this signal, we get the complex cepstrum.

$$\hat{s}(n) = \mathcal{F}^{-1}\hat{S}(j\omega) \tag{A.17}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform. This process is explained in Figure A.6 in which the complex logarithm is applied on the complex FFT which is fed forward to the IFFT block to achieve the homomorphic deconvolution.

If we drop the imaginary part of the above equation, we are left with only magnitude term, which is defined as power cepstrum of the signal. The difference between complex cepstrum and power cepstrum is that the former is reversible whereas the later is not perfectly a reversible process. A detailed guide to the cepstrum transform can be found in Childers *et al.* (1977) and Rabiner and Schafer (1978, Chap. 7). All this process does is to separate the excitation signal from the effect of vocal tract. Figure A.7(b) shows the cepstrum transform of the signal shown in Figure A.7(a). Consistent high peaks appear at regular intervals in the excitation signal and the difference between two consecutive peaks give the pitch period. Notice the rapidly decaying low-time components representing the combined effect of the vocal tract.

Appendix B

Non-Metric Dimensional Scaling Plots

The supplementary NMDS plots for the 10 folds and corresponding speakers for all of the tested databases are given here. For both Aibo-Mont and Aibo-Ohm databases, we have given the plots for only first 6 speakers for each database.

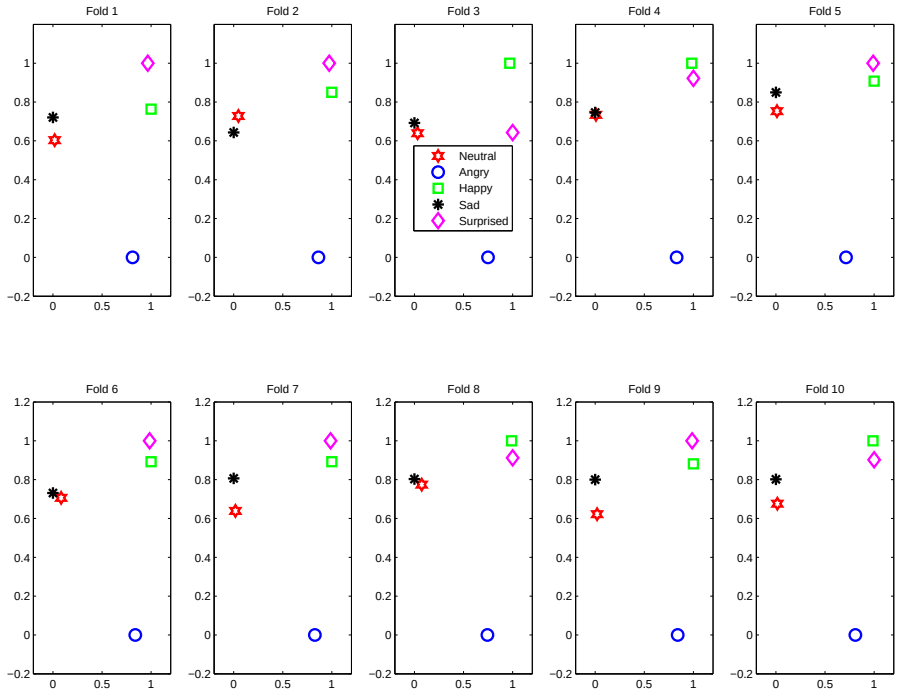


Figure B.1: NMDS plots for the 10 folds of the DES database in the speaker-dependent case.

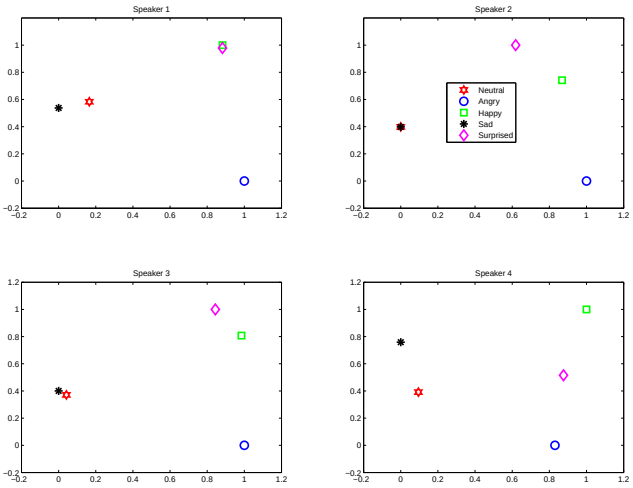


Figure B.2: NMDS plots for the 4 speakers of the DES database in the speaker-independent case.

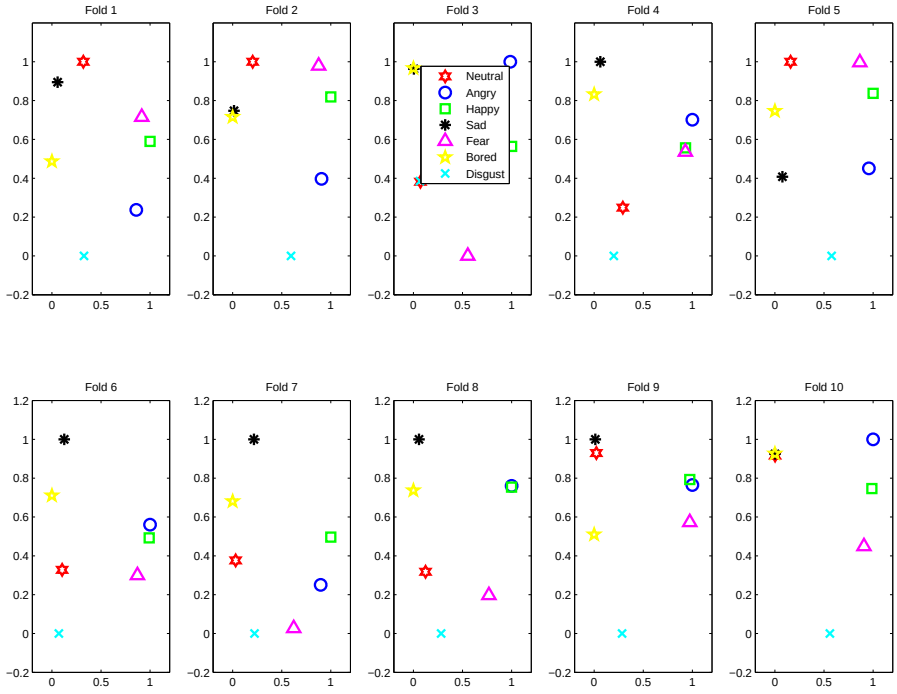


Figure B.3: NMDS plots for the 10 folds of the Berlin database in the speaker-dependent case.

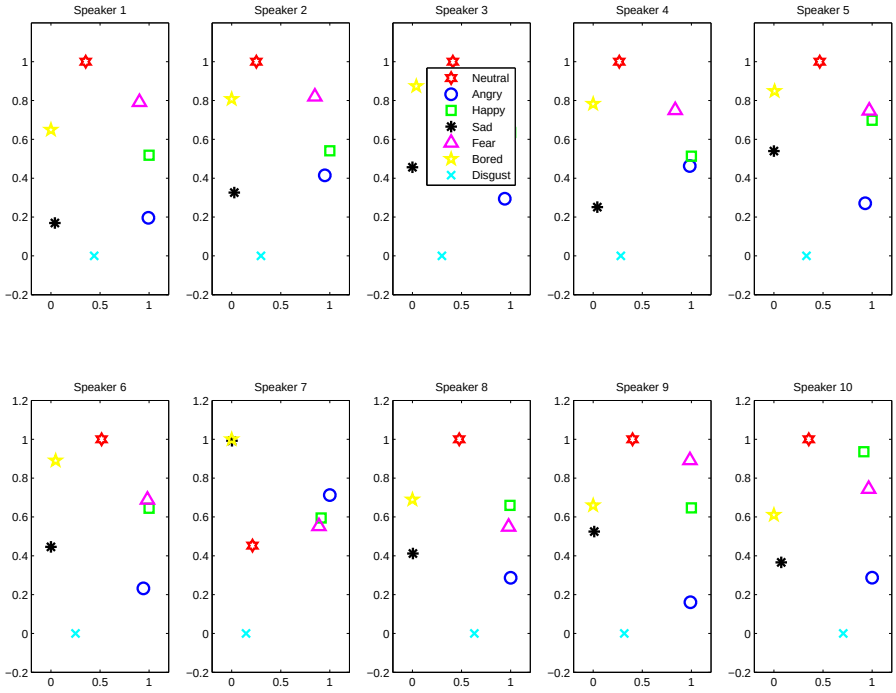


Figure B.4: NMDS plots for the 10 speakers of the Berlin database in the speaker-independent case.

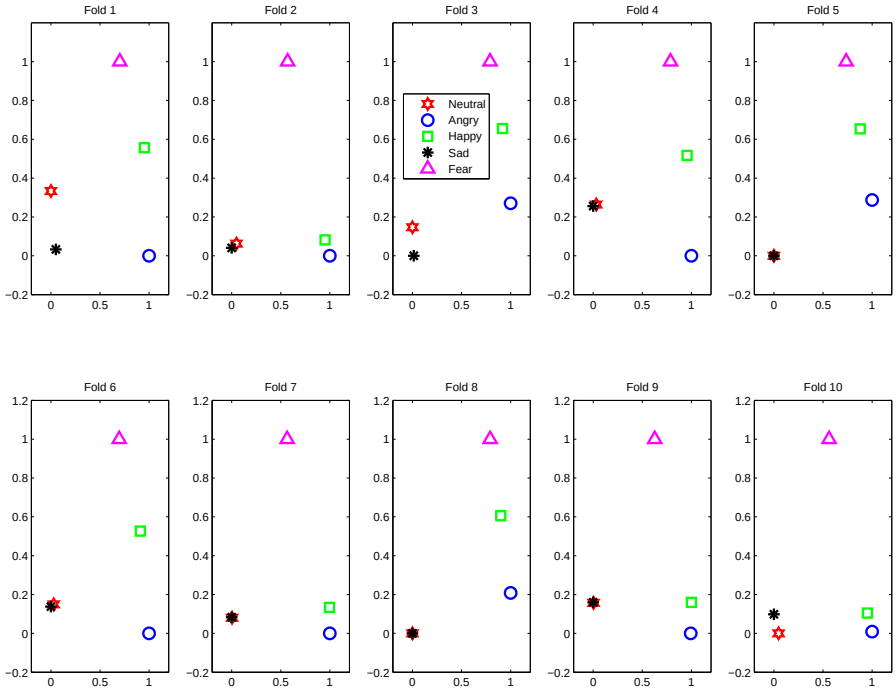


Figure B.5: NMDS plots for the 10 folds of the Serbian database in the speaker-dependent case.

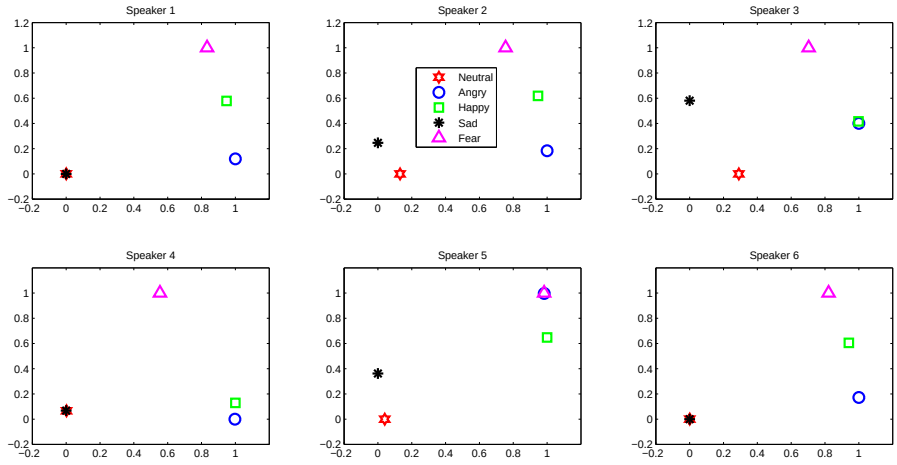


Figure B.6: NMDS plots for the 6 speakers of the Serbian database in the speaker-independent case.

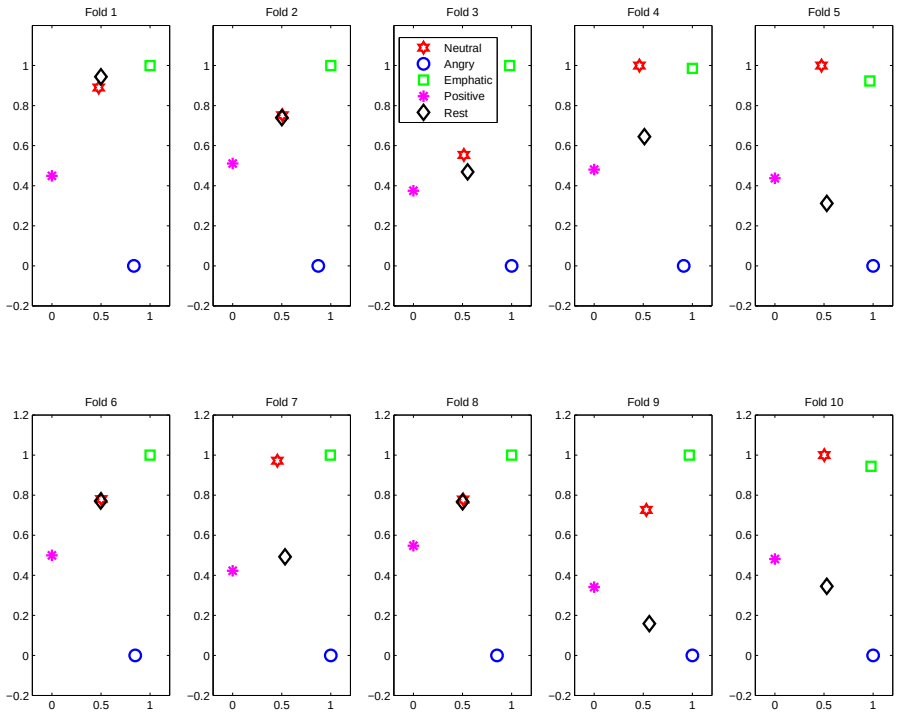


Figure B.7: NMDS plots for the 10 folds of the Aibo-Mont database in the speaker-dependent case.

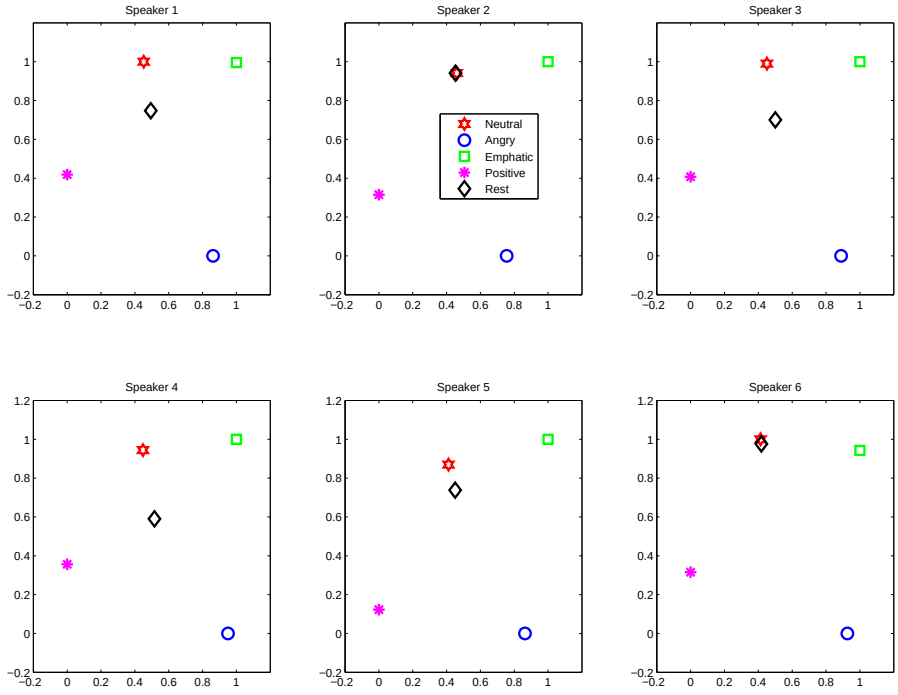


Figure B.8: NMDS plots for the first 6 speakers of the Aibo-Mont database in the speaker-independent case.

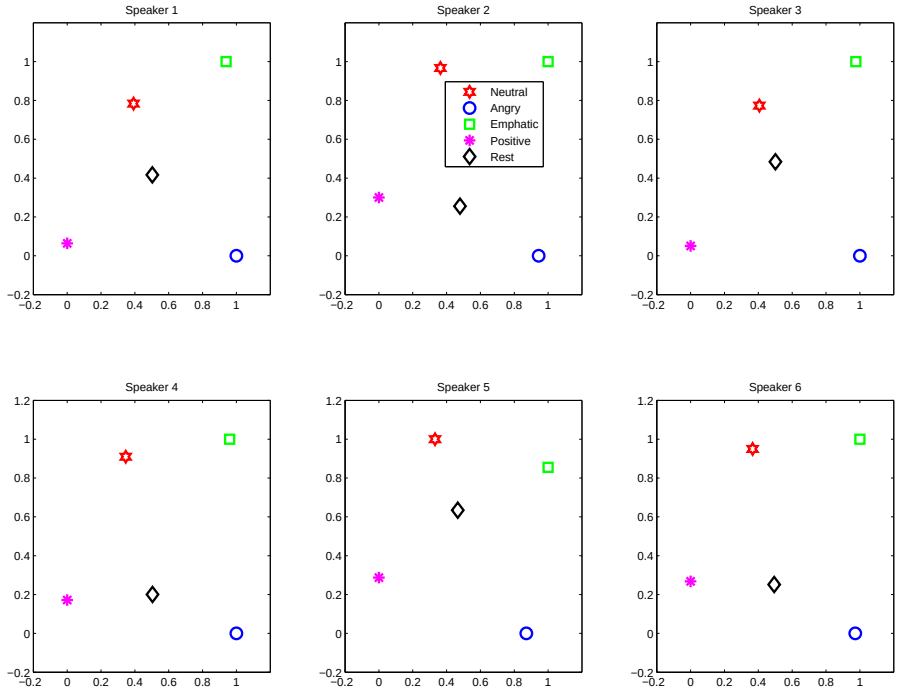


Figure B.9: NMDS plots for the first 6 speakers of the Aibo-Ohm database in the speaker-independent case.

Bibliography

Abelin, A. and Allwood, J. (2000). Cross linguistic interpretation of emotional prosody. In *Proceedings of the ISCA ITRW on Speech and Emotion*, pages 110–113, Belfast, UK.

Ai, H., Litman, D., Forbes-Riley, K., Rotaru, M., Tetreault, J., and Purandare, A. (2006). Using system and user performance features to improve emotion detection in spoken tutoring dialogues. In *Proceedings of Interspeech 2006*, pages 1682–1685, Pittsburgh, PA.

Atal, B. and Hanauer, S. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *Journal of the Acoustical Society of America*, **50**(2B), 637–655.

Atal, B. and Rabiner, L. (1976). A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **24**(3), 201–212.

Balentine, B. and Morgan, D. P. (2002). *How to Build a Speech Recognition Application: A Style Guide for Telephony Dialogs*. Enterprise Integration Group, Salem, OR, second edition.

Batliner, A., Hacker, C., Steidl, S., Noth, E., D'Arcy, S., Russell, M., and Wong, M. (2004). 'You stupid tin box' – children interacting with the AIBO robot: a cross-linguistic emotional speech corpus. In *Proceedings of 4th Language Resources and Evaluation Conference LREC, 2004*, pages 171–174, Lisbon, Portugal.

Batliner, A., Steidl, S., Schuller, B., Seppi, D., Laskowski, K., Vogt, T., Devillers, L., Vidrascu, L., Amir, N., Kessous, L., and Aharonson, V. (2006). Combining efforts for improving automatic classification of emotional user states. In *Proceedings of Information Society-Language Technologies Conference IS-LTC*, pages 240–245, Ljubljana, Slovenia.

Batliner, A., Steidl, S., Hacker, C., and Nöth, E. (2008). Private emotions versus social interaction: a data-driven approach towards analysing emotion in speech. *User Modeling and User-Adapted Interaction*, **18**(1), 175–206.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ.

Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of Institute of Phonetic Sciences, University of Amsterdam*, **17**, 97–110.

Breazeal, C. and Aryananda, L. (2002). Recognition of affective communicative intent in robot-directed speech. *Autonomous Robots*, **12**(1), 83–104.

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., and Weiss, B. (2005). A database of German emotional speech. In *Proceedings of 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 1517–1520, Lisbon, Portugal.

Campbell, W. M., Sturim, D. E., and Reynolds, D. A. (2006). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letter*, **13**(5), 308–311.

Casale, S., Russo, A., Scebba, G., and Serrano, S. (2008). Speech emotion classification using machine learning algorithms. In *IEEE International Conference on Semantic Computing, '08*, pages 158–165, Santa Clara, CA.

Chang, C. (2001). On the polynomial mixed 0-1 fractional programming problems. *European Journal of Operational Research*, **131**(1), 224–227.

Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357.

Cheveigne, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, **111**(4), 1917–1930.

Childers, D., Skinner, D., and Kemerait, R. (1977). The cepstrum: A guide to processing. *Proceedings of the IEEE*, **65**(10), 1428–1443.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. (2001). Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, **18**(1), 32–80.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK.

Darwin, C. (1859). *On the origin of species*. John Murray, London, UK.

de Borda, J. (1781). Memoire sur les elections au scrutin. *Historie de l'Académie Royale des Sciences, Paris*.

Dellaert, F., Polzin, T., and Waibel, A. (1996). Recognizing emotion in speech. In *Proceedings of Fourth International Conference on Spoken Language Processing, ICSLP'96*, pages 1970–1973, Philadelphia, PA.

Douglas-Cowie, E., Campbell, N., Cowie, R., and Roach, P. (2003). Emotional speech: Towards a new generation of databases. *Speech Communication*, **40**(2), 33–60.

Douglas-Cowie, E., Cowie, R., Sneddon, I., Cox, C., *et al.* (2007). The HUMAINE database: addressing the collection and annotation of naturalistic and induced emotional data. In *Affective Computing and Intelligent Interaction*, volume 4738 of *Lecture Notes in Computer Science*, pages 488–500. Springer, Berlin, Heidelberg.

Eide, E. and Gish, H. (1996). A parametric approach to vocal tract length normalization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'96*, pages 346–348, Atlanta, GA.

El Ayadi, M., Kamel, M., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, **44**(3), 572–587.

Engberg, I. and Hansen, A. (1996). *Documentation of the Danish Emotional Speech Database DES*. Center for PersonKommunikation, Institute of Electronic Systems, Alborg University, Aalborg, Denmark.

Eyben, F., Wöllmer, M., and Schuller, B. (2009). OpenEAR - Introducing the Munich open-source emotion and affect recognition toolkit. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction, ACII'09*, pages 1–6, Amsterdam, The Netherlands.

Eyben, F., Batliner, A., Schuller, B., Seppi, D., , and Steidl, S. (2010). Cross-corpus classification of realistic emotions – some pilot experiments. In *Proceedings of 7th International Conference on Language Resources and Evaluation, LREC'10*, pages 77–82, Valletta, Malta.

Fant, G. (1960). *Acoustic Theory of Speech Production*. Mouton, The Hague, The Netherlands.

Farran, B., Saunders, C., and Niranjan, M. (2010). Machine learning for intrusion detection: Modeling the distribution shift. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 232–237, Kittilä, Finland.

Fernandez, R. and Picard, R. W. (2005). Classical and novel discriminant features for affect recognition from speech. In *Proceedings of 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 4–8, Lisbon, Portugal.

Gales, M. and Woodland, P. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, **10**(4), 249–264.

Gouvêa, E. and Stern, R. (1997). Speaker normalization through formant-based warping of the frequency scale. In *Fifth European Conference on Speech Communication and Technology, EUROSPEECH'97*, pages 1139–1142, Rhodes, Greece.

Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, E. (2007). A kernel method for the two sample problem. In *Proceedings of Neural Information Processing Systems, NIPS'07*, pages 513–520, Vancouver, Canada.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schoelkopf, B. (2009). *Dataset Shift in Machine Learning*, chapter 8. Covariate Shift and Local Learning by Distribution Matching, pages 131–160. MIT Press, Cambridge, MA.

Hall, M. (1999). *Correlation-based Feature Selection for Machine Learning*. Ph.D. thesis, Department of Computer Science, The University of Waikato, New Zealand.

Hansen, J. and Bou-Ghazale, S. (1997). Getting started with SUSAS: A speech under simulated and actual stress database. In *Proceedings of 5th European Conference on Speech Communication and Technology, Eurospeech'97*, pages 1743–1746, Rhodes, Greece.

Haq, S. and Jackson, P. (2009). Speaker-dependent audio-visual emotion recognition. In *Proceedings of International Conference on Auditory-Visual Speech Processing (AVSP'08)*, pages 53–58, Norwich, UK.

Haq, S. and Jackson, P. (2010). *Machine Audition: Principles, Algorithms and Systems*, chapter 8. Multimodal Emotion Recognition, pages 398–423. IGI Global, Hershey, PA.

Hassan, A. and Damper, R. (2012). Classification of emotional speech using 3DEC hierarchical classifier. *Speech Communication*. In Press.

Hassan, A. and Damper, R. I. (2009). Emotion recognition from speech using extended feature selection and a simple classifier. In *Proceedings of 10th Annual of the International Speech Communication Association, Interspeech'09*, pages 2403–2406, Brighton, UK.

Hassan, A. and Damper, R. I. (2010). Multi-class and hierarchical svm's for emotion recognition. In *Proceedings of 11th Annual of the International Speech Communication Association, Interspeech'10*, pages 2354–2357, Makuhari, Japan.

Hassan, A., Damper, R., and Niranjan, M. (2012a). On acoustic emotion recognition: Compensating for covariate shift. *IEEE Transactions on Audio, Speech and Language Processing*. In Preparation.

Hassan, A., Damper, R., and Niranjan, M. (2012b). On acoustic emotion recognition: compensating for covariate shift – supplementary results. Technical report, Faculty of Physics and Applied Sciences, University of Southampton.

Hermes, D. (1988). Measurement of pitch by subharmonic summation. *Journal of the Acoustical Society of America*, **83**(1), 257–264.

Hopkins, I., Gower, M., Perez, T., Smith, D., Amthor, F., Wimsatt, F., and Biasini, F. (2011). Avatar assistant: Improving social skills in students with an ASD through a computer-based intervention. *Journal of Autism and Developmental Disorders*, **41**(11), 1543–1555.

Hozjan, V., Kacic, Z., Moreno, A., Bonafonte, A., and Nogueiras, A. (2002). Interface databases: Design and collection of a multilingual emotional speech database. In *Proceedings of 3rd International Conference on Language Resources and Evaluation, LREC'02*, pages 2024–2028, Canary Islands, Spain.

Hsu, C. and Lin, C. (2001). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13**(2), 415–425.

Hsu, C., Chang, C., and Lin, C. (2003). *A Practical Guide to Support Vector Classification*. Department of Computer Science, National Taiwan University, Taiwan.

Jones, C. and Jonsson, I. (2008). Using paralinguistic cues in speech to recognise emotions in older car drivers. In *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*, volume 4868 of *Lecture Notes in Computer Science*, pages 229–240. Springer, Berlin, Heidelberg.

Jovicic, S. T., Kacic, Z., Dordevic, M., and Rajkovic, M. (2004). Serbian emotional speech database: Design, processing and evaluation. In *Proceedings of 9th Conference on Speech and Computer, SPECOM'04*, pages 77–81, St. Petersburg, Russia.

Kabir, A., Barker, J., and Giurgiu, M. (2010). An approach to vocal tract length normalization by robust formant. In *The International Conference on Circuits, Systems and Signals*, pages 345–348, Malta.

Kanamori, T., Hido, S., and Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, **10**, 1391–1445.

Kim, S., Georgiou, P., Lee, S., and Narayanan, S. (2007). Real-time emotion detection system using speech: Multi-modal fusion of different timescale features. In *IEEE 9th Workshop on Multimedia Signal Processing, MMSP 2007*, pages 48–51, Chania, Greece.

Kruskal, J. B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, **29**(2), 115–119.

Lee, C. and Narayanan, S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, **13**(2), 293–303.

Lee, C., Yildirim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., and Narayanan, S. (2004). Emotion recognition based on phoneme classes. In *Proceedings of 8th European Conference on Speech Communication and Technology, Interspeech'04*, pages 889–892, Jeju Island, Korea.

Lee, C., Mower, E., Busso, C., Lee, S., and Narayanan, S. (2009). Emotion recognition using a hierarchical binary decision tree approach. In *Proceedings of 10th Annual of the International Speech Communication Association, Interspeech'09*, pages 320–323, Brighton, UK.

Lee, C., Mower, E., Busso, C., Lee, S., and Narayanan, S. (2011). Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, **53**(9-10), 1162–1171.

Lefter, I., Rothkrantz, L., Wiggers, P., and van Leeuwen, D. (2010). Emotion recognition from speech by combining databases and fusion of classifiers. In *Proceedings of the 13th International Conference on Text, Speech and Dialogue*, pages 353–360, Berlin, Heidelberg. Springer-Verlag.

Leggetter, C. and Woodland, P. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer speech and language*, **9**(2), 171–185.

Li, Y. and Zhao, Y. (1998). Recognizing emotions in speech using short-term and long-term features. In *Proceedings of Fifth International Conference on Spoken Language Processing, ICSLP'98*, pages 2255–2258, Sydney, Australia.

Luengo, I., Navas, E., and Hernáez, I. (2009). Combining spectral and prosodic information for emotion recognition in the Interspeech 2009 Emotion Challenge. In *Proceedings of 10th Annual of the International Speech Communication Association, Interspeech'09*, pages 332–335, Brighton, UK.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, **63**(4), 561–580.

Markel, J. (1972). The SIFT algorithm for fundamental frequency estimation. *IEEE Transactions on Audio and Electroacoustics*, **20**(5), 367–377.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of 5th European Conference on Speech Communication and Technology, Eurospeech'97*, pages 1895–1898, Rhodes, Greece.

McKeown, G., Valstar, M., Cowie, R., and Pantic, M. (2010). The SEMAINE corpus of emotionally coloured character interactions. In *IEEE International Conference on Multimedia and Expo, ICME'10*, pages 1079–1084, Singapore.

McLeod, P. and Wyvill, G. (2005). A smarter way to find pitch. In *Proceedings of International Computer Music Conference*, pages 138–141, Barcelona, Spain.

McTear, M. (2002). Spoken dialogue technology: Enabling the conversational user interface. *ACM Computing Surveys*, **34**(1), 90–169.

Mladenić, D. and Brank, J. (2004). Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th Annual International ACM SIGIR Conference, SIGIR'04*, pages 234–241.

Murray, I. and Arnott, J. (1993). Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *Journal of the Acoustical Society of America*, **93**(2), 1097–1108.

Murray, I. and Arnott, J. (2008). Applying an analysis of acted vocal emotions to improve the simulation of synthetic speech. *Computer Speech and Language*, **22**(2), 107–129.

Nguyen, H., Franke, K., and Petrovic, S. (2009). Optimizing a class of feature selection measures. In *Proceedings of Neural Information Processing Systems, NIPS, Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity and Polyhedra (DISCML)*, Vancouver, Canada.

Nogueiras, A., Moreno, A., Bonafonte, A., and Mario, J. (2001). Speech emotion recognition using hidden Markov models. In *Proceedings of 7th European Conference on Speech Communication and Technology, Eurospeech'01*, pages 2679–2682, Aalborg, Denmark.

Noll, M. A. (1967). Cepstrum pitch determination. *The Journal of the Acoustical Society of America*, **41**(2), 293–309.

Nwe, T., Foo, S., and De Silva, L. (2003). Speech emotion recognition using hidden Markov models. *Speech Communication*, **41**(4), 603–623.

O'Connor, J. and Arnold, G. (1973). *Intonation of Colloquial English*. Longmans, London, UK, second edition.

Oppenheim, A. and Schafer, R. (1968). Homomorphic analysis of speech. *IEEE Transactions on Audio and Electroacoustics*, **16**(2), 221–226.

Ortony, A. and Turner, T. (1990). What's basic about basic emotions? *Psychological Review*, **97**(3), 315–331.

Oudeyer, P.-Y. (2003). The production and recognition of emotions in speech: Features and algorithms. *International Journal of Human-Computer Studies*, **59**(1–2), 157–183.

Pan, S. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345–1359.

Pan, S., Tsang, I., Kwok, J., Yang, Q., *et al.* (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, **22**(99), 1–12.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1226–1238.

Picard, R. W. (1997). *Affective Computing*. MIT Press, Cambridge, MA.

Pitz, M. and Ney, H. (2005). Vocal tract normalization equals linear transformation in cepstral space. *IEEE Transactions on Speech and Audio Processing*, **13**(5), 930–944.

Planet, S., Iriondo, I., Socoró, J., Monzo, C., and Adell, J. (2009). GTM-URL contribution to the Interspeech 2009 Emotion Challenge. In *Proceedings of 10th Annual of the International Speech Communication Association, Interspeech'09*, pages 316–319, Brighton, UK.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research.

Platt, J., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In *Proceedings of Neural Information Processing Systems, NIPS'99*, pages 547–553, Denver, CO.

Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, **15**(11), 1119–1125.

Quast, H. (2002). *Automatic Recognition of Nonverbal Speech: An Approach to Model the Perception of Para- and Extralinguistic Vocal Communication with Neural Networks*. Master's thesis, Machine Perception Lab, Institute for Neural Computation, University of California, San Diego, CA.

Rabiner, L. R. and Sambur, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell Systems Technical Journal*, **54**(2), 297–315.

Rabiner, L. R. and Schafer, R. W. (1978). *Digital Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, NJ.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, **3**, 1357–1370.

Ramanan, A., Suppharangsan, S., and Niranjan, M. (2007). Unbalanced Decision Trees for multi-class classification. In *International Conference on Industrial and Information Systems, ICIIS'07*, pages 291–294, Sri Lanka. IEEE.

Rong, J., Li, G., and Chen, Y. (2009). Acoustic feature selection for automatic emotion recognition from speech. *Information Processing and Management*, **45**(3), 315–328.

Rosenblatt, M. (1963). *Proceedings of the Symposium on Time Series Analysis*. John Wiley and Sons, New York, NY.

Ross, M. J., Shaffer, H. L., Cohen, A., Freudberg, R., and . Manley, H. J. (1974). Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, **22**(5), 353 – 362.

Russell, J. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, **39**(6), 1161–1178.

Russell, J., Lewicka, M., and Niit, T. (1989). A cross-cultural study of a circumplex model of affect. *Journal of Personality and Social Psychology*, **57**(5), 848–856.

Scherer, K. (2000). *The Neuropsychology of Emotion*, chapter 6. Psychological models of emotion, pages 137–162. Oxford University Press, Oxford/New York.

Scherer, K. (2003). Vocal communication of emotion: A review of research paradigms. *Speech Communication*, **40**(1-2), 227–256.

Schiel, F., Steininger, S., and Türk, U. (2002). The SmartKom multimodal corpus at BAS. In *Proceedings of the 3rd Language Resources and Evaluation Conference, LREC'02*, pages 200–206, Canary Islands, Spain.

Schuller, B., Rigoll, G., and Lang, M. (2004). Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'04*, pages 577–580, Montreal, Quebec, Canada.

Schuller, B., Muller, R., Al-Hames, M., Lang, M., and Rigoll, G. (2005a). Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. In *Proceedings of 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 805–808, Lisbon, Portugal.

Schuller, B., Reiter, S., Muller, R., Al-Hames, M., Lang, M., and Rigoll, G. (2005b). Speaker independent speech emotion recognition by ensemble classification. In *IEEE International Conference on Multimedia and Expo, ICME'05*, pages 864–867, Amsterdam, The Netherlands.

Schuller, B., Rigoll, G., Grimm, M., Kroschel, K., Moosmayr, T., and Ruske, G. (2007a). Effects of in-car noise-conditions on the recognition of emotion within speech. In *Proceedings of the 33rd Annual Conference on Acoustics, DAGA'07*, pages 305–306, Stuttgart, Germany.

Schuller, B., Batliner, A., Seppi, D., Steidl, S., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Amir, N., Kessous, L., *et al.* (2007b). The relevance of feature type for the automatic classification of emotional user states: Low level descriptors and functionals. In *Proceedings of International Conference on Spoken Language Processing, Interspeech'07*, pages 2253–2256, Antwerp, Belgium.

Schuller, B., Vlasenko, B., Eyben, F., Rigoll, G., and Wendemuth, A. (2009a). Acoustic emotion recognition: A benchmark comparison of performances. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2009*, pages 552–557, Merano, Italy.

Schuller, B., Steidl, S., and Batliner, A. (2009b). The Interspeech 2009 Emotion Challenge. In *Proceedings of 10th Annual of the International Speech Communication Association, Interspeech'09*, pages 312–315, Brighton, UK.

Schuller, B., Vlasenko, B., Eyben, F., Wöllmer, M., Stuhlsatz, A., Wendemuth, A., and Rigoll, G. (2010). Cross-corpus acoustic emotion recognition: Variances and strategies. *IEEE Transactions on Affective Computing*, **1**(2), 119–131.

Schuller, B., Valstar, M., Eyben, F., McKeown, G., Cowie, R., and Pantic, M. (2011a). AVEC 2011–The first international audio/visual emotion challenge. In *Proceedings of the 4th international conference on Affective Computing and Intelligent Interaction - Volume Part II*, pages 415–424, Berlin, Heidelberg.

Schuller, B., Zhang, Z., Weninger, F., and Rigoll, G. (2011b). Using multiple databases for training in emotion recognition: To unite or to vote? In *Proceedings of 12th Annual of the International Speech Communication Association, Interspeech'11*, pages 1553–1556, Florence, Italy.

Shaffer, H. L., Ross, M. J., and Cohen, A. (1973). AMDF pitch extractor. *Journal of the Acoustical Society of America*, **54**(1), 340–340.

Shahid, S., Krahmer, E., and Swerts, M. (2008). Real vs. acted emotional speech: Comparing Caucasian and South Asian speakers and observers. In *Proceedings of the 4th International Conference on Speech Prosody*, pages 669–672, Campinas, Brazil.

Shami, M. and Verhelst, W. (2007). An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech. *Speech Communication*, **49**(3), 201–212.

Shaukat, A. and Chen, K. (2008). Towards automatic emotional state categorisation from speech signals. In *Proceedings of 9th Annual of the International Speech Communication Association, Interspeech'08*, pages 2771–2774, Brisbane, Australia.

Sidorova, J. (2009). Speech emotion recognition with TGI+.2 classifier. In *Proceedings of the European Chapter of the Association for Computational Linguistics, EACL'09 Student Research Workshop*, pages 54–60, Athens, Greece.

Siegel, S. and Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York, NY, second edition.

Slaney, M. and McRoberts, G. (2003). BabyEars: A recognition system for affective vocalizations. *Speech Communication*, **39**(3–4), 367–384.

Snell, R. and Milinazzo, F. (1993). Formant location from LPC analysis data. *IEEE Transactions on Speech and Audio Processing*, **1**(2), 129–134.

Steidl, S. (2009). *Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech*. Ph.D. thesis, Department of Computer Science, University Erlangen-Nuremberg, Erlangen, Germany.

Sugiyama, M., Blankertz, B., Krauledat, M., Dornhege, G., and Muller, K. (2006). Importance-weighted cross-validation for covariate shift. In *28th Annual Symposium of the German Association for Pattern Recognition*, pages 354–363, Berlin, Germany.

Tickle, A. (2000). English and japanese speakers' emotion vocalisation and recognition: A comparison highlighting vowel quality. In *Proceedings of the ISCA Workshop on Speech and Emotion*, pages 104–109, Newcastle, Northern Ireland.

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. (2008). Direct density ratio estimation for large-scale covariate shift adaptation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 443–454, Atlanta, Georgia.

Van der Maaten, L., Postma, E., and Van den Herik, H. (2009). Dimensionality reduction: A comparative review. Technical report, Tilburg University, Netherlands.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York, N.Y.

Ververidis, D. and Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech Communication*, **48**(9), 1162–1181.

Vidrascu, L. and Devillers, L. (2005). Detection of real-life emotions in call centers. In *Proceedings of 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 1841–1844, Lisbon, Portugal.

Vijayakumar, S. and Wu, S. (1999). Sequential support vector classifiers and regression. In *International Conference on Soft Computing*, pages 610–619, Genoa, Italy.

Vogt, T. and André, E. (2005). Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition. In *IEEE International Conference on Multimedia and Expo, ICME'05*, pages 474–477, Amsterdam, The Netherlands.

Wilting, J., Krahmer, E., and Swerts, M. (2006). Real vs. acted emotional speech. In *Proceedings of International Conference on Spoken Language Processing, Interspeech'06*, pages 1093–1097, Pittsburgh, PA.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, second edition.

Xiao, Z., Dellandrea, E., Dou, W., and Chen, L. (2010). Multi-stage classification of emotional speech motivated by a dimensional emotion model. *Multimedia Tools and Applications*, **46**, 119–145.

Yang, B. and Lugger, M. (2010). Emotion recognition from speech signals using new harmony features. *Signal Processing*, **90**(5), 1415–1423.

Yildirim, S., Narayanan, S., and Potamianos, A. (2011). Detecting emotional state of a child in a conversational computer game. *Computer Speech and Language*, **25**(1), 29–44.