Evaluation of different Support Vector Machines (SVM) for Speaker Identification

Rouhana Jhumka

A thesis submitted to the Department of Electrical Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Cape Town, June 2004

Declaration

I declare that this project report is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signed by candidate

Cape Town
30 June 2004

Abstract

This study is an investigation into four support vector machines (SVM) kernels. SVMs have gained much acceptance in classification tasks since their inception in the 1990s. The central feature of SVM is the implicit mapping of input data to some higher-dimensional feature space. This is achieved through the use of kernel functions. Popular kernel functions include gaussian, polynomial, sigmoid and linear. This list is by no means exhaustive. The work done in this thesis compares the four kernels mentioned.

Attaining maximum performance with SVM requires optimizing the hyperparameters that are embedded in the kernel function. The results obtained from the experiments performed indicate that the linear kernel's performance was the worst compared to the other three kernels. This can be attributed to the fact that the hyperplane separating the classes of data is not linear. Moreover, it was shown that all the other three kernels achieved relatively the same performance for each data set considered. We can also conjecture from the results that the gaussian kernel took excessive time to converge. This fact is also reported in [52].

SVM was then applied in a hybrid GMM/SVM system using the optimized hyperparameters of each kernel function. The gaussian SVM kernel provided the best performance at the expense of computational time. The identification error rate using the hybrid system was further reduced by 7.7%.

Acknowledgements

First of all, I would like to thank ALLAH (SWT) for the strength and courage He gave me throughout this study.

I am very thankful to my supervisor, Dr D. Mashao, who provided excellent guidance and support throughout this thesis. He was always available when I needed assistance and was very keen to help me with my problems and assist with supplying software codes. I would like to thank him whole-heartedly.

This thesis would not have been possible without the constant support and encouragement provided by my parents and family. They have given much love and incentive so that I can reach my goals. I will forever be grateful for their unconditional love and the faith they have in me.

I would also like to thank Khalil who was always there for me. He provided me with constant moral support. He was always there to listen to my frustations when experiments failed. He encouraged me to strive for excellence.

A big thanks goes to uncle Razack, aunty Fatima and the rest of the family. They have been very supportive and helpful. A special thanks goes to aunty Fatima for the succulent meals she provided me.

I would like to express my most sincere gratitude to Bilal and Sam. You have contributed in innumerable ways to the completion of this thesis.

I would also like to thank Deon Kallis who always enquired about my work and was always willing to assist me.

Finally, I owe a debt of gratitude to Lerato, Limpho and Ofentse for their willingness to help with problems I encountered while typing my thesis.

Contents

List of Figures

List of Tables

List of Symbols

ASR	—	Automatic Speech Recognition
ERM	—	Empirical Risk Minimization
HMM		Hidden Markov Models
GMM	—	Gaussian Mixture Models
$LPCC$	—	Linear Prediction Cepstral Coefficients
NN	—	Neural Networks
SID	—	Speaker Identification
SRM	—	Structural Risk Minimization
SVM	—	Support Vector Machines
PFS	—	Parameterised Feature Sets
$MFCC$	—	Mel-frequency Cepstral Coefficients
VQ	—	Vector Quantization

Chapter 1

Introduction

1.1 Background

Communication through words to convey feelings, desires, emotions is natural to most of us. It is this ease of relating from one person to another that engendered the possibility of computers to capture the essence of speech to perform certain tasks, e.g issuing commands to a computer instead of using the mouse or the keyboard. Typing on a computer can be a daunting task for someone who is not familiar with a computer. Hence that person will feel more comfortable having a speech interface to the computer than typing.

Speech recognition is still an active research area. The fast-growing demand for human-computer interaction has warranted better machine performance. However, automatic speech recognition stills falls short of human performance. In digit recognition, machines are at least two orders of magnitude worse than humans and on conversational noisy speech, humans are an order of magnitude higher than machines [41].

Speech recognition further generated speaker recognition systems. Speaker recognition can be classified into two categories, namely identification and verification [18, 76]. In speaker identification, the system establishes the identity of the speaker from a set of speakers while speaker verification sees the authentication of speaker based on who he/she claims to be.

However, speaker identification systems are far from duplicating the human ear. On clean speech, 100% accuracy is achieved [123, 45, 98] but on noisy speech, the identification system performs poorly. The highest identification rate to date reported by [98] is approximately 72% on noisy data.

An SID system can be broken down into two stages: a front-end stage and a back-end stage. The front-end part, also known as the feature extraction process, deals with the conversion of an acoustic signal into a series of features that contain relevant information. The characteristics obtained from the front-end stage are known as feature vectors and they are speaker-dependent. These feature vectors are then fed into the back-end stage, also known as the classification part. The aim of this stage is to match an unknown speaker's features with the stored ones and make a decision based on the maximum scores.

1.2 Support Vector Machines (SVM)

Support Vector Machines have been applied successfully on different classification problems and they have been shown to perform better than other classifiers, e.g neural networks (NN) [116]. SVMs are state-of-the-art pattern recognition techniques that stem from statistical learning theory [139, 140]. They are based on guaranteed risk bounds of statistical learning theory. They use the principle of structural risk minimization (SRM) which minimizes the empirical risk on the test data yielding good generalization.

SVM is inherently a binary classifier and it finds the optimal linear hyperplane separating two classes of data. This optimal hyperplane is found by maximizing the distance between the two classes. SVM treats the classification problem as a quadratic optimization problem. For non-linear decision boundaries, SVM uses the kernel trick [2] to transform input data to a higher dimensional space and construct the linear hyperplane in that feature space. It then projects the data back to the input space where a non-linear decision boundary is obtained. No *a priori* knowledge is needed for the mapping from input space to feature space. The kernel function performing the mapping must be symmetric and satisfy Mercer's theorems [29, 139, 140]. The kernel functions allow SVM to be flexible in searching for various feature spaces.

1.3 Objectives of this thesis

The objectives of this thesis are to:

- study various SVM kernels.
- search for the optimal hyperparameters of the SVM kernels using the NTIMIT database.

- implement a hybrid GMM/SVM system.
- report on the performance of the kernels and on the hybrid system.
- draw necessary conclusions and make recommendations.

1.4 Contribution

The primary objective of this thesis is to explore the use of different SVM kernels for speaker recognition using the NTIMIT database. SVMs have been applied in the past few years on a wide range of classification problems [13, 70, 129, 104, 106]. This thesis investigates the performance of four different kernels applied on the NTIMIT [108] database. Instead of comparing SVM with other classifiers, it is important to investigate SVM itself by using its different kernels. Some studies comparing SVM kernels can be found in [40, 138, 126, 143]. Furthermore, this thesis also looks, to a lesser degree, at a GMM/SVM hybrid system by using the optimal hyperparameters found by the evaluation of the various kernels. The main focus of attention in this thesis is the evaluation of the SVM kernels rather than on the hybrid system.

1.5 Previous work

Experiments were performed at the beginning of this research to investigate the performance of SVM as compared to GMM when applied to a speaker identification task [69]. The comparative study was done using thirty-eight people from the NTIMIT database. It was found that SVM outperforms GMM for limited data. As the amount of training data increases, GMM achieves better performance than SVM. In [95], Mashao proposed an *N*- best hybrid GMM/SVM system that takes advantage of the fact that SVM performs better on less data. He reported that the complementary system reduced the identification rate by 11%. Mashao used the gaussian SVM kernel. This initiated the investigation of SVM kernels amongst themselves to find out the best hyperparameters for the various kernels. Furthermore, the optimal hyperparameters obtained were then applied to the complementary system developed in [95].

1.6 Structure of thesis

Chapter 2 gives an overview of current speech technology. It also provides background information on speaker recognition, more specifically speaker identification. Some applications of SID are also highlighted in this chapter.

Chapter 3 provides information on support vector machines (SVM). This chapter includes detailed information about the choice of kernels that is central to the task in this thesis. Four kernels are investigated, namely gaussian, sigmoid, polynomial and linear.

Chapter 4 deals with the experimentation part of this research. Each SVM kernel is investigated to obtain the best hyperparameters for the task. Results together with the relevant discussions are given for three sets of data used in each case.

Chapter 5 deals with a GMM/SVM hybrid system. Each kernel, dealt with in chapter 4, produced a set of optimal parameters. They are in turn applied to the hybrid system.

Chapter 6 concludes this thesis based on the results obtained in chapters 4 and 5. Further work pertaining to the task dealt with in this report is also included in this chapter.

Chapter 2

Speech Technology

The aim of this chapter is to give an overview about speech technology. The problem of speaker identification and its applications are also dealt with.

2.1 Introduction

What is speech? If we are fortunate to have the sense of sound, then we hear and listen to different kinds of sounds everyday. One of them is speech. Let us pause for one moment and try to understand the real meaning of speech. There are several definitions of speech. Speech can be defined as "communication by word of mouth" [148], "making definite vocal sounds that form words to express thoughts and ideas" [66] and "the oral medium of transmission for language" [43], amongst others. Basically, speech is "making definite vocal sounds that form words to express thoughts and ideas" [66]. It is one of the most important ways of human communication.

The powerful advances in technology have made it possible to have computer-human interaction. Speech recognition is one of the cutting edge technologies using the computer as its medium. It allows the user to have more control when using a computer. For example, instead of using the mouse for performing certain commands, the user can issue commands to the computer by speaking into a microphone connected to the computer. Thus it converts speech into text. Speech recognition can be divided into three main categories [33]:

- discrete speech and continous speech.

Discrete speech recognition requires the user to say one word at a time, thus requiring a pause after each word. On the other hand, continuous speech recognition allows the user to speak sentences without pauses between words.

- speaker-dependent and speaker-independent

  Speaker-dependent recognition means that the system is trained for a particular voice. Every single user must train the system to recognize his/her voice. In other words, speaker-dependent systems are tailored to suit the speaker. Speaker-independent systems are designed such that any speaker can train the system by uttering words to the computer. The system is not trained on the voice of the speaker but rather on what the speaker is saying.

- small vocabulary and large vocabulary

  Speech recognition systems depend also on the vocabulary size. The latter still presents a challenge to an automatic speech recognition system. Vocabulary size depends largely on the type of application that speech recognition will be used on. For example, an office dictation programme might require 10000 words while an industrial inspection task needs only 200 words [33]. Large vocabulary usually ranges from 2000 to 200000 words while small vocabulary ranges from 2 to 200 words.

Three types of errors can be made by a speech recognizer:

- rejection error

  A rejection error occurs when the machine does not classify the utterance. It simply rejects it instead. A simple solution to the problem is to repeat the utterance again.

- substitution error

  This type of error occurs when the recognizer mistakes the speaker's utterance and substitutes it with another word. This misrecognition is not distinguishable from a legal utterance. Hence, one can verify that the user's utterance was understood.

- insertion error

With this type of error, the recognizer classifies noise as a legal utterance. For example, if there were other people in the room talking or there was music playing or even a simple sneeze, the recognizer interprets the background noise as a valid word.

2.2 Brief history of speech recognition[1]

Automatic speech recognition (ASR) has been a challenge for many decades. The ultimate goal of an ASR machine is to enable the machine to recognize a spoken word and understand its meaning. It will be an added boon if the machine can understand any word from any speaker. Attempts at ASR date as far back as the 1930s. In 1936, AT&T's Bell Labs produced the first electronic speech synthesizer (text-to-speech) called the Voder. In 1952, [34] built an isolated digit recognition for one speaker. Olson and Belar [109] tried to recognize ten syllables of a single speaker in 1956.

The 1960s saw a dramatic change in methodology that contributed positively to the ASR field. Martin et al. [92] developed a set of methods that considerably reduced the variability of recognition scores. Vintsyuk [141] proposed the concepts of dynamic time warping (DTW). Research done by [117] in the field of continuous speech recognition was a major contribution to the ASR field. From the 1970s till today, there have been rapid development of ASR. From speaker-dependent ASR, researchers at AT&T's Bell Labs started experiments that were speaker-independent [34]. In the early 1970s a new modelling method, namely hidden markov models (HMM), was proposed by Lenny Baum of Princeton University. In 1971, DARPA (Defense Advanced Research Projects Agency) set up a programme to understand continuous speech. Hence, there was great impetus in developing ASR systems for continuous speech as opposed to isolated word/digit recognition.

ASR has been evolving quite rapidly since then with newer and better methods proposed to achieve high accuracy. Tremendous advances have been made in the past years. In 1982, Dragon Systems was founded. In 1995, they released the first dictation speech recognition system to consumers. Charles Schwab is the first company to allocate resources to develop a speech recognition IVR (Interactive Voice Responce) system with Nuance. The system handles up to 50,000 requests daily and is found to be 95% accurate. Dragon Systems introduced "Naturally Speaking", which is the first continuous speech dictation software, in 1997.

[1] Source taken from [114, 147, 5]

In 2000, TellMe introduced the first voice portal which uses speech technology to interpret the speaker's commands and text-to-speech technology to present the information. The year 2000 also saw the launch of Office Depot's first voice-enabled internet applications by NetBytel.

2.3 Applications of speech technology[2]

Speech recognition systems are used in several areas in our daily lives. The opportunities of speech recognition are immeasurable. Dictation packages exist that allow people to input text to a computer without the use of a keyboard. The voice portal has brought about the voice browsing technology. The latter is very advantageous as it allows people without computers to get online by using the telephone. Airline schedules, weather forecast, TV programmes and stock quotes can be accessed using the voice browsing hands-free technology. The latter is also useful for people with disabilities such as visual impairment.

Speech recognition can also be used at call centres. British Telecom offers a fully speech-enabled White Pages Directory Assistance system. Cellphone companies are also making use of speech recognition. It is possible to utter the name of the person you want to call and the call is made. A further cellphone application is the implementation of recording memos, speeches, etc and then have a computer convert the speech to text. Another application of speech technology is in the sending of emails. Users are also allowed to send email over the phone. SpeechMail is one such system that allows this application.

2.4 Speech production

Speech production starts with the conceptualization of an idea that a speaker wants to transmit to the listener [114]. The speaker then converts the idea into a set of words based on the speaker's language. Finally, the speaker executes neuromuscular commands causing the vocal organs to move accordingly, thus producing a sequence of sounds.

The vocal tract consists of a set of organs for speech production. The vocal tract is the most important speech production system. It consists of the laryngeal pharynx, oral pharynx, oral cavity, nasal pharynx and nasal cavity [18]. The main three cavities in the vocal tract are the throat, oral and nasal cavities. The vocal tract starts at the vocal folds and

[2]Source taken from [147]

ends at the lips. The reader is refered to [114] for a schematic drawing of the vocal tract. A male's vocal tract is approximately 17 cm long [114].

The lungs are the source of air for speech production. The airflow from the lungs causes the vocal folds to vibrate. The velum (soft palate) controls the airflow to the nasal cavity. The tongue, mouth, velum, jaw, teeth and lips, known as *articulators*, produce different sounds based on their positions [74]. During exhalation, the vocal folds can either be tensed or relaxed [114]. The vibratory motion of tensed vocal folds produces a *voiced* sound. For relaxed vocal folds, an *unvoiced* sound is produced by the passage of air through a narrow constriction in the vocal tract.

The vocal tract carries speaker-dependent characteristics [36, 64, 114, 18] and hence can be used for a speaker identification system. Another distinguishing factor of speech that can be used to identify speakers is the *learned* characteristic which includes speaking rate, dialect and prosodic effects [18].

2.5 Background to speaker recognition

For centuries, humans have been performing speaker recognition in their daily lives. The identification of voices start at a very early age, e.g babies recognize their mother's voice. Speaker identification is quite easy for the human ear. But automatically identifying a person remains a challenging task. In order to reliably identify a person, one needs to know certain distinguishing characteristics of that person such as the face, voice, iris and fingerprint. This is known as biometric person recognition [78]. The latter is one of the three ways of authenticating a person [72]. The other two processes of automatic person recognition involve information a person has, e.g a pin number or password and a token the person possesses e.g a credit card. Biometric person recognition is the most effective amongst the three methods as credit cards can be stolen and pin numbers forgotten.

2.5.1 Speaker identification

Speaker recognition can be classified into speaker verification and speaker identification. Speaker verification is the process whereby a speaker is authenticated on who he/she claims to be. This involves a binary decision on the validity of the claim. Speaker identification, on the other hand, is the process of recognizing an unknown speaker out of a database of speakers given a speech utterance as input. In this case, there are N possible

outcomes compared to just two possible outcomes in the verification case.

SID systems can be either closed set or open set. In a closed set speaker identification system, all the test utterances belong to one of the enrolled speakers [18] whereas an open set allows test utterances from unknown speakers [53]. Speaker identification systems can further be subdivided into two categories based on text dependence. Text dependent systems require the speaker to utter specific words or sentences. The linguistic content of the utterance is of utmost importance. Text independent systems, on the other hand, require no prior knowledge of the speaker's utterance. Figure 2.1 shows a general speaker recognition problem. The speaker verification task follows the same taxonomy as the SID task.

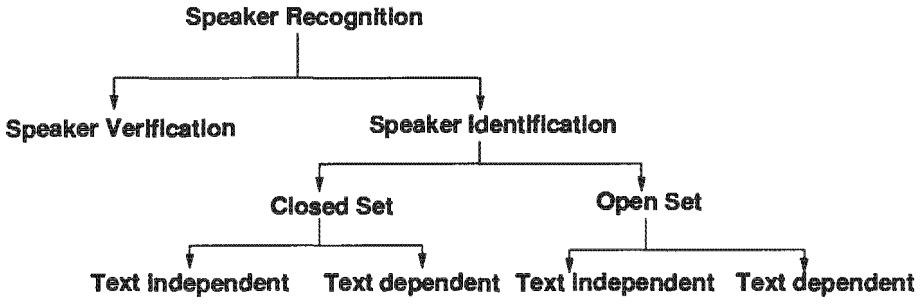

Figure 2.1: Speaker Recognition Problem

2.5.2 The paradigm for speaker identification

Speech conveys various kinds of information about the speaker including gender, age, dialect, emotional state and attitude [145] amongst others. There are three main processes that happen so that speech is possible [114]. Firstly, a person formulates an idea. Then, he/she chooses the right words/sentences according to his/her language and finally the neuro-muscular commands from the brain move the vocal production organs accordingly.

The voice characteristics of a speaker are largely determined by the size and shape of the vocal tract [114]. The size of the male vocal tract is more than that of females and children. Females and children have smaller vocal folds than males. As a result, their overall pitch is higher. Females have pitch ranging from 120-200 Hz while males's pitch frequency vary between 60-120 Hz [112]. It is important in a SID task to extract those discriminative characteristics for modelling.

2.5.2.1 Feature extraction

We can consider a speech signal as a sequence of features with speaker-dependent characteristics. A *feature extraction* process separates each speaker according to these characteristics. The front-end of an SID system is the feature extraction process. The latter converts the speech signal into a sequence feature vectors. It is important that sufficient information is extracted from the speech signal for effective modelling [54]. Feature extraction can be considered as a process that reduces the amount of data from the original speech signal while retaining speaker-dependent information [74].

Various feature extraction methods can be found in the literature. These include linear prediction cepstral coefficients (LPCC) [114, 90, 134], mel-frequency cepstral coefficients (MFCC) [133], parameterized feature sets (PFS) [93], etc. These methods consist of the front-end part of an SID system.

2.5.2.2 Classification

The *classification* stage is a decision-making process whereby the identity of a speaker is determined based on stored information. The feature vectors extracted from each speaker utterance are fed to the classification stage. A model of each speaker is generated from these feature vectors. To identify a speaker, the feature vectors extracted from the unknown speech signal are matched to the models of each enrolled speaker and a decision is made as to the identity of the speaker.

The back-end of a SID sytem is known as the *classification* stage. Back-ends used in SID system are vector quantization (VQ) [132, 75], gaussian mixture models (GMM) [121], neural networks (NN) [116], hidden markov models (HMM) [26], support vector machines (SVM) [47, 32], etc. GMM is the most used classification system and it has been shown to produce the best results combined with PFS as the front-end [98]. Recently, SVM learning engine has emerged as a powerful classification system.

Figure 2.2 shows a general structure of an SID system and figure 2.3 shows the back-end stage. This study uses PFS as the front-end and SVM as the back-end. The main focus of this thesis is on evaluating the different SVM kernels. A detailed discussion of SVM and its kernels are given in chapter 3.

Figure 2.2: Block Diagram of a Speaker identification system

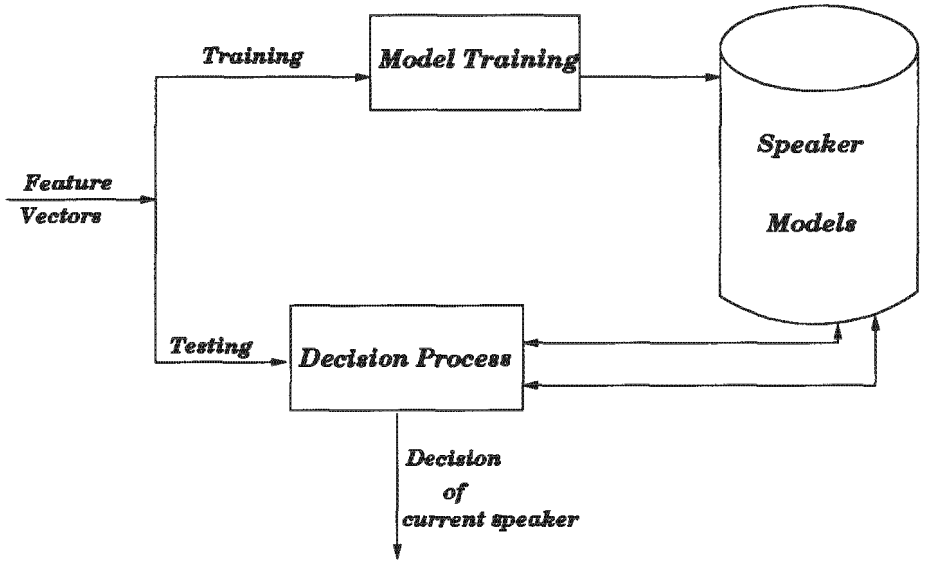

Figure 2.3: Classification stage of SID

2.5.3 Applications of SID

The main applications of SID range from security systems to forensics. In forensic applications [125], prerecorded voice samples or telephonic recordings of the perpetrator's voice can be used to determine whether the suspect is the perpetrator. This application of SID is very useful for criminal investigations. Forensics is one of the earliest real-world applications of SID.

Security applications can use SID for access control to buildings, computer networks, e-banking and phone transactions. These systems need authorization to gain access to them. Hence a person can be identified according to his biometric, voice, before entering a building or accessing a computer. In the case of e-banking [10], a person's information can be stored on a smart card. Whenever, he/she needs to perform e-banking transactions, he/she inserts the card into a device attached to the computer and identifies himself/herself by speaking into a microphone. A person can also make changes to his/her accounts or even shop over the phone by identifying themselves.

2.6 Summary

This chapter has provided background information on speech technology. Speaker identification, which is a branch of speech recognition, was also discussed. Finally, this chapter discussed some of the potential applications of SID.

Chapter 3

Support Vector Machines (SVM)

The aim of this chapter is to provide an understanding of the basics underlying support vector machines. The chapter will start by considering the empirical risk minimization principle compared to structural risk minimization principle. Support vector classification for linear and non-linear cases are then described. Finally, the importance of the choice of kernel is dealt with.

3.1 Introduction

Support Vector Machines (SVMs) are machine learning systems that were originally introduced by Vapnik [139] and co-workers. SVMs are discriminative classifiers that have good generalization characteristics.

SVMs have been successfully applied to many real-world problems. The latter include, but are not limited to, the automatic categorization of gene expression data from DNA microarrays [13], machine vision [110], text categorization [70], hand-written digit recognition [83, 82], colour-based classification [21], phonetic classification [25], speaker identification [129], time-series prediction [104, 106], musical instrument classification [91], classification of milk based on the odour and fat content of the milk [14] and the detection of microcalcifications in mammograms [42].

A Support Vector Machine (SVM) is a supervised learning technique that learns the decision surface through discrimination. In supervised learning, the algorithm tries to learn the relationship between the input and output data. The functional relationship mapping the input to the output is given by the target function, also known as the decision function

in a classification example [32].

3.2 Generalisation theory

The performance of a learning machine is given by its generalization. The latter is the ability of the learning machine to classify unseen data.

3.2.1 Empirical Risk Minimization (ERM)

Let a two-class classification problem be defined as (x_1, y_1), (x_2, y_2), ... , (x_n, y_n) where x_i , $(i = 1, \ldots , n)$ are input data and $y_i = \{+1, -1\}$ are the class labels for each data point. The aim of the learning machine is to find a classifier with the decision function, $f(x)$, such that $y = f(x)$. Let $P(x, y)$ be a probability distribution generating training and testing data [15]. The quality of the function f can be measured by minimizing the expected error as shown in equation 3.1.

$$R(f) = \int L(f(x), y) dP(x, y) \tag{3.1}$$

where L denotes an appropriately chosen loss function.

The risk function (also referred to as the actual risk) given in equation 3.1 cannot be minimized directly as no a priori information is known about the probability distribution. Hence, instead of minimizing the actual risk, the empirical risk can be minimized. The empirical risk is given by equation 3.2

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) \tag{3.2}$$

and is defined as the mean error computed over available training data. Minimization of equation 3.2 is called *Empirical Risk Minimization (ERM).* ERM is widely used in optimization algorithms for machine learning. Using ERM alone can lead to overfitting of data as the learning algorithm does not take into consideration the complexity of the learning machine.

3.2.2 Structural Risk Minimization (SRM)

Structural Risk Minimization (SRM) seeks the tradeoff between empirical risk and the capacity of the learning machine. In other words, it seeks to minimize the upper bound on the generalization error [139]. The VC (Vapnik-Chervonenkis) theory provides a way of controlling the complexity of a function class. One of the core concepts of statistical learning theory is the VC dimension. The latter is a measure of the capacity of a learning algorithm. It shows how well a learning machine can classify data.

The loss function described in equation 3.1 can only take the values 0 and 1. It can take take several values if for some η, $0 \leq \eta \leq 1$. Then for losses taking these values, the inequality bounding the risk is given by equation 3.3.

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{h(log(\frac{2n}{h}+1) - log(\frac{\eta}{4}))}{n}} \tag{3.3}$$

The parameter, h, in equation 3.3 is a non-negative integer and is called the VC dimension. If the complexity of the function class F, that f is chosen from, is restricted, then the VC dimension can also be defined as the maximum number of training points that can be shattered using functions of the class.

The second term of equation 3.3 is called the VC confidence and it depends on the chosen class of functions. *SRM* chooses the function class that minimizes the bound on the actual risk. This is achieved by training each subset of the function class so that the empirical risk is minimized. In other words, the *SRM* principle chooses the function that minimizes the right hand side of equation 3.3 [15]. Figure 3.1 is a representation of *SRM*. The optimal point on the graph describes the bound on the expected risk.

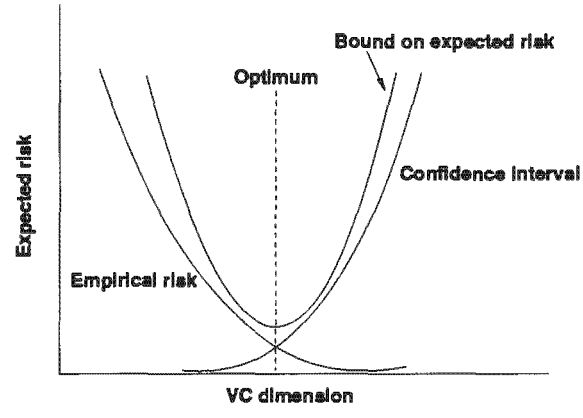

Figure 3.1: Representation of *SRM*

3.3 Training

3.3.1 Linear SVM

3.3.1.1 The separable case

Restricting the classification problem to a binary one, the goal is to separate the two classes by constructing a linear decision boundary or hyperplane. Consider the set of training samples $x_1, x_2, ..., x_m$ where $x_i \in R^n$. Each sample has a corresponding label y_1, $y_2, ..., y_m$ where $y_i \in \{-1,1\}$ indicating which of the two classes the samples belong to. Figure 3.2 shows such an arrangement of samples where the square samples denote the positive samples and the round samples denote the negative examples.

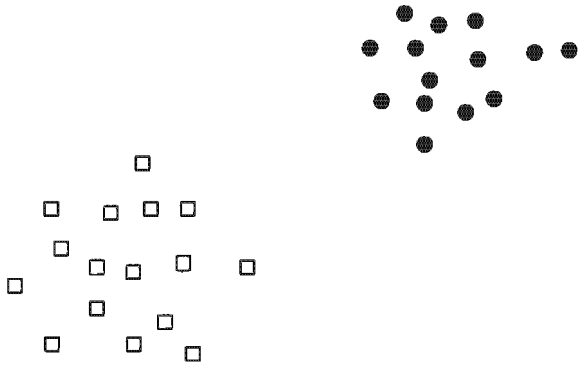

Figure 3.2: Binary representation of classification problem

Let $f(x)$ be a linear function represented by

$$f(x) = \mathbf{w}.\mathbf{x} + b \tag{3.4}$$

where $\mathbf{w}$ is a weight vector and b is a constant.

$\mathbf{w}$ and b are the parameters that control the function and the mapping that can separate the two classes is given by the sign$(f(x))$.

3.3.1.2 Optimal separating hyperplane

The pair $(\mathbf{w},b)$ defining the hyperplane between the two classes is not unique. Hence, there exists an infinite number of decision boundaries which satisfy the equation $f(x) = sign(\mathbf{w}.\mathbf{x}+b)$. The best decision boundary is the one that maximizes the margin between

the two classes while minimizing the empirical risk. Figure 3.3 shows a binary problem where the samples are linearly separable. There are different hyperplanes, for instance H1, H2, H3, H4 that will classify the samples perfectly but H3 is the one that maximizes the margin between the classes as shown in figure 3.4. SRM, thus, provides the optimal hyperplane.

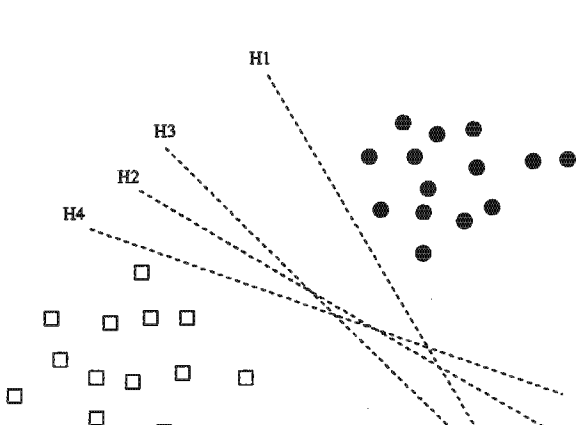

Figure 3.3: Representation of various hyperplanes

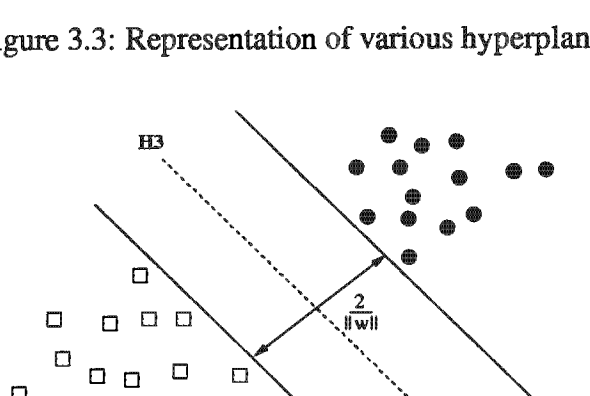

Figure 3.4: Hyperplane with maximum margin

The points that lie on the hyperplane separating the data satisfy the following equation:

$$\mathbf{w}.\mathbf{x} + b = 0 \tag{3.5}$$

The hyperplane separates the data if and only if

$$(\mathbf{w}.\mathbf{x} + b) > 0 \tag{3.6}$$

for $y_i = 1$ and

$$(\mathbf{w}.\mathbf{x} + b) < 0 \tag{3.7}$$

for $y_i = -1$

Scaling $\mathbf{w}$ and b leads to equations 3.8 and 3.9.

$$(\mathbf{w}.\mathbf{x} + b) \geq 1 \tag{3.8}$$

for $y_i = 1$

$$(\mathbf{w}.\mathbf{x} + b) \leq -1 \tag{3.9}$$

for $y_i = -1$

Combining equations 3.8 and 3.9 leads to equation 3.10.

$$(\mathbf{x}_i.\mathbf{w} + b)\mathbf{y}_i \geq 1 \forall i \tag{3.10}$$

The optimal hyperplane is obtained by finding the plane that maximizes the distance between the hyperplance and the closest data point. The distance to the closest data sample is given by equation ?

$$d = min_{\{\mathbf{x}_i|\mathbf{y}_i=1\}} \frac{\mathbf{w}.\mathbf{x}_i + b}{\|\mathbf{w}\|} - max_{\{\mathbf{x}_i|\mathbf{y}_i=-1\}} \frac{\mathbf{w}.\mathbf{x}_i + b}{\|\mathbf{w}\|} \tag{3.11}$$

Equation 3.10 shows that the minimum and maximum values are ± 1. Hence, equation 3.11 is simplified to obtain equation 3.12.

$$d = \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{3.12}$$

The margin associated with the optimal hyperplane is thus equal to $\frac{2}{\|w\|}$ as shown in figure 3.4.

The theory of Lagrange multipliers [8] can be used to solve the optimization problem. Introducing a Lagrange multiplier, α_i , for each constraint, the primal form of the function is:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i(y_i(x_i.w + b) - 1), \alpha_i \geq 0 \tag{3.13}$$

The optimization of equation 3.13 leads to a convex quadratic programming problem. Minimizing the Lagrangian leads to the following equations:

$$\sum_i \alpha_i y_i = 0 \tag{3.14}$$

and

$$w = \sum \alpha_i y_i x_i \tag{3.15}$$

Substituting equations 3.14 and 3.15 into equation 3.13 yields the dual optimization formulation:

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i.x_j \tag{3.16}$$

Adjusting the Lagrange multipliers, α, maximizes the objective function.

From equations 3.15 and 3.4, we can see that the decision function can be defined as:

$$f(\mathbf{x}) = \sum \alpha_i y_i(\mathbf{x}_i.\mathbf{x}) + b \tag{3.17}$$

By using the Karush-Kuhn-Tucker conditions [80], we obtain the following equation:

$$\alpha_i[y_i(x_i.w + b) - 1] = 0 \tag{3.18}$$

All training data with non-zero α's are known as support vectors. All other training data have α_i equal to zero. Support vectors define the decision boundary. They carry all the information about the given problem. α_i is non-zero for all examples satisfying:

$$y_i(x_i.w + b) = 1 \tag{3.19}$$

3.3.1.3 The non-separable case

The above sections dealt with the case where the training data are completely separable. Many classification problems often involve non-separable training data. Given such problems, the goal is still to find the linear boundary that maximizes the margin and minimizes the errors on the training data. The optimization equation 3.13 no longer applies as it will not converge. To allow for misclassification errors and penalize them accordingly, a *soft margin* [28] is introduced. This results in the introduction of slack variables, ξ, and a penalty function, C.

The resulting constrained minimization problem is

$$L = \frac{1}{2}\|w\|^2 + C\sum_i \theta_\sigma(\xi_i), \sigma > 0 \tag{3.20}$$

subject to

$$(x_i.w + b)y_i \geq 1 - \xi_i \qquad i = 1, ..., l$$
$$\xi_i \geq 0 \; \forall i$$

where ξ_i are the slack variables

C is the penalty for a training error

θ is the cost function that penalizes errors

Figure 3.5 shows a classifier with training errors. Similar to the separable case, the dual optimization is given by equation 3.21.

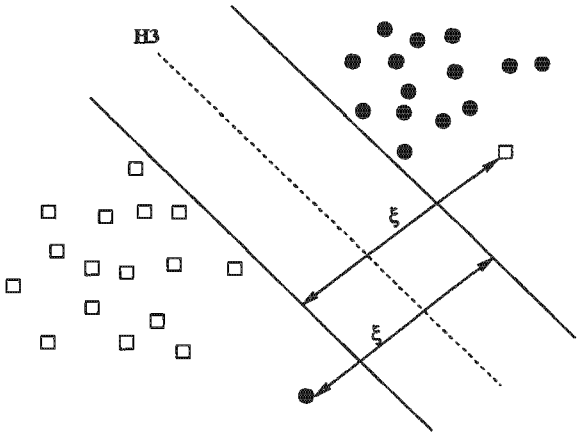

Figure 3.5: Allowing misclassification errors in non-separable case

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i . x_j \tag{3.21}$$

subject to the constraints

$$0 \leq \alpha_i \leq C$$
$$\sum_i \alpha_i y_i = 0$$

The dual optimization formulation for the non-separable case is identical to the separable case, except for an upper bound imposed on the Lagrange multipliers.

3.3.2 Non-linear SVM

There are various real-world classification problems where linear decision boundaries fail, e.g for noisy training data. Hence the need for non-linear decision boundaries arises.

SVM uses the "kernel trick" [2] whereby input data are projected to a higher dimensional space where linear separability is possible. Hence the problem is solved in the resulting space and projected back to the input space where a non-linear decision boundary is obtained.

The input vectors in all the optimization functions in equations 3.16 and 3.21 occur as dot products. Finding the optimal decision boundary directly in a higher dimensional space can be computationally expensive and very complicated. By using [2], the authors showed that the "kernel trick" can be used to achieve the mapping. With the choice of a suitable kernel, the data can be separated in the feature space despite being non-separable in input space. Let Φ be a mapping from input space to a higher dimensional feature space.

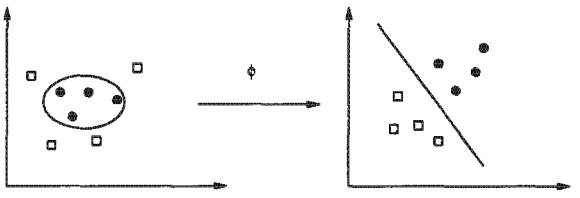

Figure 3.6: Kernel mapping from input space to feature space

$$\Phi : R^n \rightarrow R^N \tag{3.22}$$

where N is the dimensionality of the new feature space. Figure 3.6 shows such a mapping.

The dot product in input space is now replaced by $\Phi(x_i).\Phi(x_j)$. All the dot products in equations 3.16 and 3.21 can be replaced by a kernel function, K, where

$$K(x_i, x_j) = \Phi(x_i).\Phi(x_j)$$

The new optimization problem now becomes:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.23}$$

subject to the constraints $0 \leq \alpha_i \leq c$ and $\sum_i \alpha_i y_i = 0$

To classify the data, the new decision function is

$$f(x) = \sum \alpha_i y_i K(x_i, x) + b \tag{3.24}$$

and classification of unseen examples is achieved by $sign(f(x))$.

An example of a projection from input space to feature space is given below. Let the data be in a two-dimensional space $\Re^2$ and the transformation is to a $\Re^3$ space [52]. Hence, we can choose a polynomial transformation of degree 2. The kernel function can be represented as $K(x, y) = (x.y)^2$. If the mapping is defined as shown in equation 3.25,

$$\Phi(x).\Phi(y) = (x.y)^2 \tag{3.25}$$

where $x = (x_1, x_2)$ and $y = (y_1, y_2)$

then, the projection is represented by equation 3.26.

$$K(x, y) = (x_1^2, x_1 x_2, x_2 x_1, x_2^2)(y_1^2, y_1 y_2, y_2 y_1, y_2^2)^T \tag{3.26}$$

The function Φ can thus be defined as shown in equation 3.27.

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} \tag{3.27}$$

Provided the kernel function is a valid one, no knowledge about the transformation Φ is required. All eligible kernels must satisfy two conditions:

- it must be symmetric and
- it must satisfy Mercer's theorem [29, 139] which states that there exists a mapping Φ and a kernel function $K(x, y) = \Phi(x).\Phi(y)$, if and only if, for any function $g(x)$ such that

  $\int g(x)^2 dx$ is finite,
  the kernel function must satisfy the inequality:
  $\int K(x, y)g(x)g(y)dxdy > 0$

Generally if the Kernel function is represented by a dot product in some feature space, it can be used. Some of the most commonly used kernel functions are given in table 3.1. Each kernel defines a different type of feature space, for example an SVM with RBF kernels, the resulting architecture is an RBF network [16]. However, good generalization should be achieved considering the bounds on generalization error [139, 140, 32]. It is not imperative to know the functional form of the mapping Φ, since it is implicitly defined by the choice of the kernel, $K(x, y)$ [16].

Table 3.1: Different types of kernel functions

Type of Classifier	**Kernel Function, $K(x, y)$**
Linear	$(x.y + 1)$
Radial Basis Function (RBF)	$e^{-\frac{\|x-y\|^2}{2\sigma^2}}$
Polynomial	$(s\,x.y + r)^d$
Sigmoid	$\tanh(s\,x.y + r)$

3.3.3 Multiclass SVM

SVM is inherently a binary classifier. But it can be extended to handle multiple classes. Several different schemes have been proposed to handle multiclass classification problems [51, 99, 56, 60]. The *one-against-all classifier* and *one-against-one classifier* are two of the widely used classifier algorithms.

3.3.3.1 One-against-all classifier

This classification scheme constructs k SVM models where k is the number of classes. The ith SVM is trained where SVM attempts to construct a decision hyperplane separating

the ith class from the rest. Hence, all examples in the ith class are assigned postitive labels and all other examples, negative labels. Classification of an unseen test example, x, is given by the class that maximizes the decision function given by equation 3.24. The resulting classifier is :

$$\text{class of x} \equiv argmax_{i=1,.....,k}(\alpha_i y_i K(x_i, x) + b)$$

3.3.3.2 One-against-one

This classification algorithm was introduced in [77] and first used in [51] and [79]. It splits a multiclass problem into several binary problems. Hence, $k(k-1)/2$ classifiers are built whereby each one is trained from the binary set of data. After all $k(k-1)/2$ models are trained, a voting strategy [51] is used to classify unseen data. Each binary classifier assigns a vote to one of the two classes. A test sample, x, is classified as belonging to the class with the most votes. Mayoraz and Alpaydin [99] used this type of classifier.

3.4 Choice of Kernels

From the previous section, it can be seen that kernels are used for the mapping of training data from the input feature space to a higher dimensional space. An SVM's performance is highly dependent on the choice of the kernel.

The kernels given in table 3.1 are discussed in this section. These kernels differ in the VC dimension, their smoothing properties and their generalization performance [65]. Each kernel depends on various parameters. One of these parameters, C, does not appear in the non-linear mapping from input space to feature space. But it is also treated as a kernel parameter.

3.4.1 Previous work on kernel selection

Boser et al. [11] applied SVM on handwritten digit recognition problem. They used the polynomial kernel with degrees 1 to 6 and the gaussian kernels for σ between 0.1 and 4.0. They reported that maximum classification was obtained using the polynomial kernel from degree 3. In [126], SVMs are applied to phoneme classification. He used the linear, polynomial and gaussian kernels and reported that the gaussian kernel performed better. Ayat et al. [6] applied SVM to digit image recognition. They compared three

types of kernels, KMOD (kernel with moderate decreasing), polynomial and RBF. They reported that KMOD kernel achieved the highest performance but the polynomial kernel of degree 4 performed better than the RBF kernel. SVM was used in [138] to identify top quark events and no significant difference in performance was seen when using the RBF and sigmoid kernels. Dumas [40] showed that the sigmoid kernel achived the highest accuracy when SVM was applied to the classification of emotion on human faces. Wan and Campbell [143] applied SVM to speaker verification and identification tasks and [129] implemented SVM on a speaker identification system only. Wan and Campbell [143] reported that the polynomial kernel performed better than the RBF.

3.4.2 Gaussian (or RBF) kernels

The gaussian kernel function, also known as the radial basis function (RBF) kernel, is given by $K(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$. It determines the dot product in an infinite dimensional space. The weight of the kernel function decreases exponentially with the square of the distance. The width σ, which is the variance, determines the weight of near and far samples. Points that are far away from the centre of the gaussian kernel are considered to be irrelevant. The support vector lies at the centre of the gaussian with variance, σ.

The gaussian kernel has two parameters namely, σ and the cost factor, C. The latter is the trade-off between the training error and model complexity. It is also regarded as one of the hyperparameters of the gaussian kernel. The RBF kernel is the most widely used kernel for classification. It provides better generalization than the other kernels.

3.4.3 Polynomial kernels

The polynomial kernel is given by $(s\,x.y + r)^d$. This kernel maps the data into $(n + d)$ dimensional space where n is the dimensionality of the input space. The resulting decision boundary is a polynomial [16]. This kernel is less widely used than the RBF kernel because it becomes increasingly difficult to train the SVM as the value of the output increases. The output of the polynomial kernel function depends on the direction of two vectors in low-dimensional space [24]. All vectors in the same direction have a high output.

3.4.4 Sigmoid kernels

The sigmoid kernel is given by $\tanh(s\,x.y + r)$. It has three tunable hyperparameters, namely r, s and the cost function, C. The sigmoid kernel is quite popular due to its origin from neural networks [116]. Since it is not postive semi-definite (PSD), it cannot be regarded as a valid kernel. However, the kernel matrix is conditionally positive definite (CPD) for certain parameters and is therefore a valid kernel for those parameters only [87]. Optimizing the sigmoid kernel entails finding the centre of the function. s is viewed as a scaling parameter of the input data [87]. It scales the width of the sigmoid kernel. Small s values generate wide sigmoid function, while a narrow function is obtained with large s values. The r parameter shifts the sigmoid function along the x-axis. It does not affect the scaling of the sigmoid. Lin and Lin [87] showed the best performance for different ranges of s and r. Table 3.2 shows those results. The sigmoid kernel behaves like the RBF kernel for certain values of s and r.

Table 3.2: Behaviour of sigmoid kernel for a combination of hyperparameters [87]

s	r	**Results**
$+$	$-$	K is CPD after r is small; similar to RBF for small σ
$+$	$+$	in general not as good as the $(+,-)$ case
$-$	$+$	objective value $-\infty$ after r is large enough
$-$	$-$	easily an objective value of $-\infty$

3.4.5 Linear kernels

The linear kernel is the simplest kernel function. It is given by the following equation $(x.y + 1)$. It has only one tunable parameter which is the penalty term, C. The linear kernel results in the same objective function given by equation 3.21. It is a specific case of the polynomial kernel with degree 1. The linear kernel has the disadvantage that it does not perform well when the relation between the target values and attributes is non-linear. In [73], Keerthi and Lin show that the linear kernel is a special case of the RBF kernel as the linear kernel with a penalty parameter C has the same performance as the RBF kernel with parameters (C, σ).

3.5 Summary

This chapter has introduced and described the theory of support vector machines. The principles of empirical risk minimization and its relationship to structural risk minimization were also discussed. The power of SVM to map input data to a higher dimensional space was also dealt with. In addition, SVM extended to the multiclass situation was highlighted. Examples of previous work that involved a comparison of SVM kernels were provided. Finally, the choice of the kernels was discussed.

Chapter 4

Evaluations of the kernels

This chapter deals with the evaluation of SVMs using different kernels on a speaker identification problem. The NTIMIT database [108] was used for the experiments. NTIMIT is a corpus of continuous speech passed through telephone channels, hence introducing noise. The database consists of 630 speakers from eight major dialect regions of the United States. Each person in the database reads ten sentences comprising of two standard sentences (sa), five phonetically rich sentences (si) and three unique sentences (sx).

The front-end part of the SID system was not dealt with in this thesis. The front-end feature system, PFS [93], was used to obtain the feature vectors of each speaker. Three sets of two people were used for the experiments. Set A consists of two male speakers, set B is made up of two female speakers and set C comprises of a male and a female speakers. The feature vectors of each utterance consists of 30 dimensions. The leave-one-out (loo) scores of each person were then computed. The loo scores are the accuracy rates of each speaker as the hyperparameters are varied.

For the speaker identification task, a training set and a testing set were created whereby the utterances in the test set were unknown to the training phase of the SVM. The training set was made up of eight utterances and the remaining two utterances were used for testing.

4.1 Toolkits used

The experiments were performed using two different toolkits, namely SVMTorch [27] and LIBSVM [20]. The classification methods differ for the two toolkits. SVMTorch uses "one-against-all" classification while LIBSVM uses "one-against-one". The work done in

this thesis is not a comparison of the two toolkits. Hence, the results do not indicate which toolkit performed better. The results are taken from the toolkit that performed better for any particular task. Some studies that made use of the SVMTorch toolkit are [126], [101] and [19]. Other studies that used the LIBSVM toolkit are [138], [40] and [63].

4.2 Data sets used for evaluations

Three sets of two people were randomly chosen and used for the experiments. The three letters of each speaker represents the speaker's initials and the number differentiates the speakers with identical initials. Set A consists of two male speakers taken from dialect regions dr2 (Northern dialect) and dr6 (New York City dialect). The speaker chosen from dr2 was 'mdm2' and that from dr6 was 'dsc0'. Set B is made up of two female speakers namely 'mem0' and 'sxa0'. Speaker 'mem0' comes from dialect region dr1 (New England dialect) while speaker 'sxa0' originates from dialect region dr7 (Western dialect). Set C comprises of a male and a female speakers. The male speaker chosen was 'zmb0' from dialect region dr2 (Northern dialect). The female speaker was 'adg0' from dialect region dr4 (South Midland dialect).

4.3 Comparison of different kernels

Four different kernels were used to examine their performance compared to one another. They are namely, gaussian, sigmoid, polynomial and linear kernels.

4.3.1 Effect of kernel parameters on performance

The performance of the SVM machine depends on the hyperparameters of the kernels used. Since we do not know a priori the optimal hyperparameters needed to ensure high accuracy, a search for these parameters is needed. We would like to find the parameters that would give the best generalization performance. The data are scaled between +1 and -1. Scaling of the data is performed to avoid numerical problems during calculation. Another advantage of scaling is to avoid attributes in greater numeric ranges to dominate those in smaller numeric ranges. As mentioned earlier, three sets of two speakers were chosen to perform the experiments. It would have been more advantageous if the whole database (630 speakers) was used for the selection of best parameters, but due to excessive

computational time, it does not seem feasible to do so.

4.3.1.1 Using a gaussian kernel

The gaussian kernel function is given by $K(x,y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$. It has two parameters namely the variance of the kernel, σ and the cost factor, C, used to penalize training errors during training. A search was performed to obtain the optimal hyperparameters. [73] suggests taking $\log C$ and $\log\sigma^2$ as the parameters of the hyperparameter space. [94] showed that the optimal parameters for unscaled data are $C = 2^{15}$ and $\sigma = 2^{15}$. Experiments were done to check the validity of [94] and the results are reported in table 4.1. Figure 4.1 gives some visual information for the same range of parameters. The results clearly show that the performance of the classifier is dependent on the parameters. From table 4.1, we can see that the performance does not change dramatically for increasing values of C. This shows that a larger value of C does not affect the accuracy of the system as when $C = 2^{10}$, all the data have been accounted for; the optimal solution has been reached. The blank spaces in table 4.1 are due to the fact that the SVM machine was not able to converge at the time of completion of this thesis.

The data of the three sets of speakers mentioned earlier were then scaled and a "grid search" was performed. The data was scaled between -1 and 1 by using equation 4.1.

$$x_1 = \frac{2(x - min)}{max - min} - 1 \tag{4.1}$$

where min and max denote the minimum and maximum values of the ith attribute respectively.

A logarithmic scale for the parameters was still used. [95] reported hyperparameter dependence on data scaling. Experiments done in this thesis found that the best parameters for all three sets were found to be $\sigma = 1$ and $C = 256$. The same parameters were found in [95]. Irrespective of gender, the best parameters for the gaussian kernel are the same. Figures 4.2, 4.3 and 4.4 show the accuracies for the three sets. We can deduce that the system does not confuse between gender and between the identities of female speakers. For male speakers, however, there is a 14.45% identification error. The fact that for gender identification, the system performed better is supported in [86, 1]. Slaney [131] also reported that classifying female speakers is easier than male speakers.

In [95], Mashao showed a higher accuracy using a different type of scaling as shown in equation 4.2.

$$x_1 = \frac{x}{max(abs(x))} \tag{4.2}$$

The latter was applied to check the validity of [95]. The accuracy of set A improved to 91.11%. Figure 4.5 shows the accuracy for set A using scaling defined in equation 4.2.

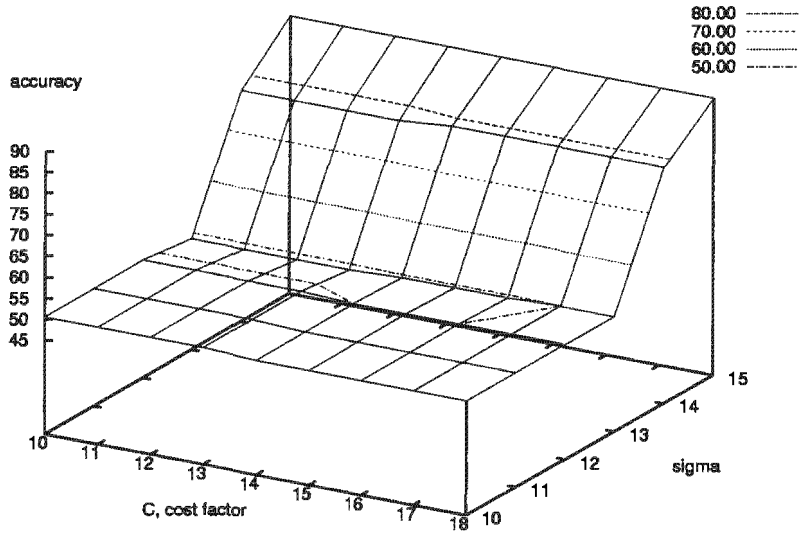

Figure 4.1: Performance on unscaled data

Table 4.1: Effect of kernel parameters on the classification performance of RBF kernel

$logC$ **for** $\sigma = 2^{15}$	**Accuracy, %**	$log\sigma$ **for** $C = 2^{13}$	**Accuracy, %**
10	88.88	10	50.56
11	88.88	11	50.56
12	88.88	12	50.56
13	88.88	13	48.88
14	88.76	14	77.77
15	88.76	15	88.88
16	88.76	16	-
17	88.88	17	-
18	88.76	18	-

4.3.1.2 Using a sigmoid kernel

The sigmoid kernel is given by $\tanh(s\,x.y + r)$. It has three tunable hyperparameters, namely r, s and the cost function, C. Since it is not known which r, s and C are the best for our problem, a parameter search is done. As the number of parameters increases,

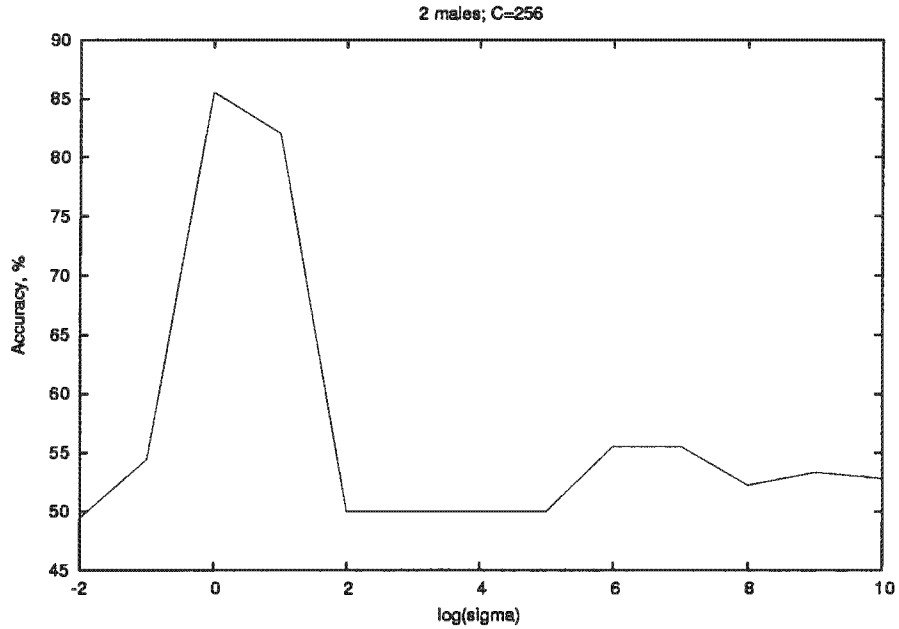

Figure 4.2: Hyperparameter dependence for set A

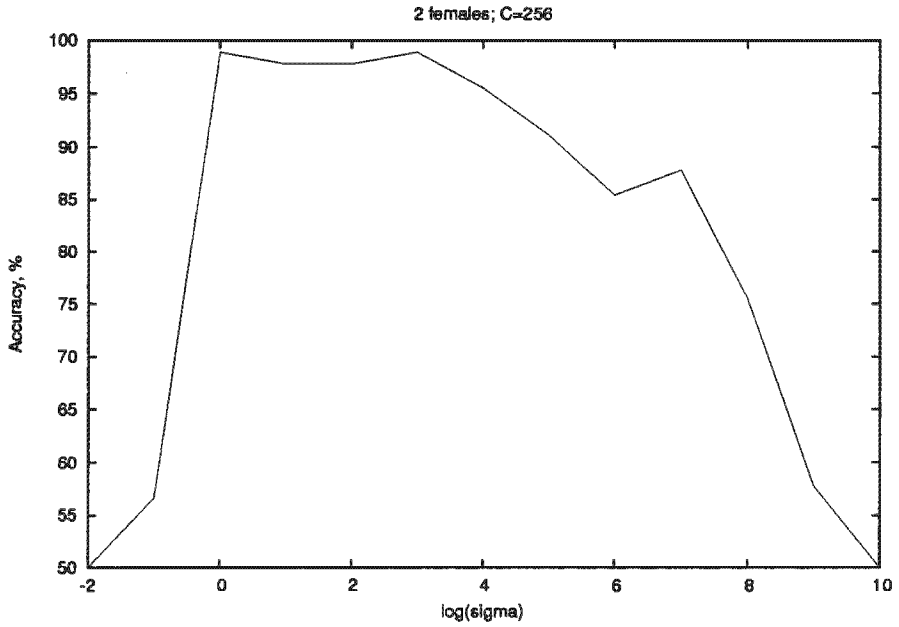

Figure 4.3: Hyperparameter dependence for set B

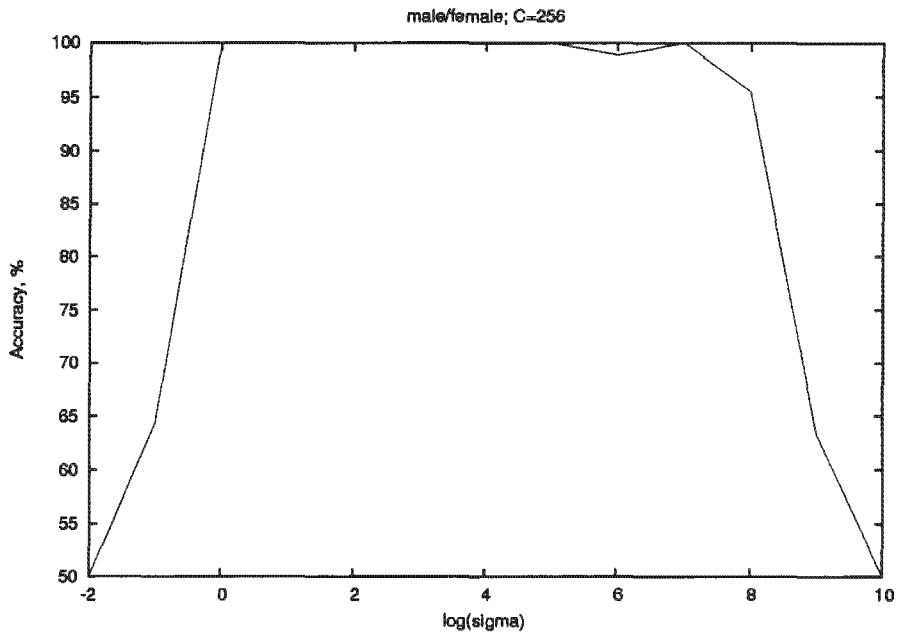

Figure 4.4: Hyperparameter dependence for set C

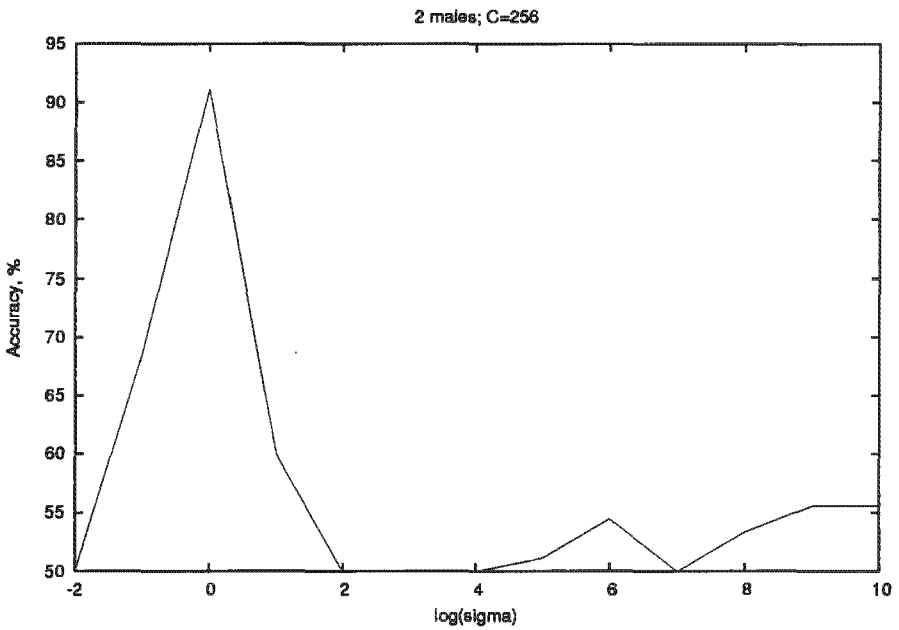

Figure 4.5: Hyperparameter dependence for set A using scaling described in equation 4.2

the search for optimal hyperparameters becomes increasingly difficult. Lin and Lin [87] showed that $s > 0$ and $r < 0$ are suitable for the sigmoid kernel. Hence, we used these conclusions to perform the search.

For set B, a grid search was conducted on different r, s and $log_2(C)$. This search is represented in figure 4.6. From figure 4.6, a better region was identified and a finer grid search was then performed. Figure 4.7 shows a refined grid with s set at 0.11 and the seach was performed on $log_2(-r)$ from 1 to 1.5 and $log_2(C)$ from -2 to 10. The best (r, s, c) are $(-2.8, 0.11, 1024)$ with an accuracy of 98.88%.

The same principle was used for grid searches for sets A and C. For set C, $s = 0.11$ was found to be the optimal as the case was for set B. Figure 4.8 shows a coarse grid for set C and figure 4.9 shows its refined version. The best (r, s, c) are $(-2.8, 0.11, 1024)$ with a 100% accuracy.

For set A however, the best s was found at 8.05. Figure 4.11 shows a coarse grid search and figure 4.12 shows its corresponding refined version. The best (r, s, c) are $(-40, 8.05, 1.15)$ with an accuracy of 72.22%. This means that for the male speakers, the sigmoid function fitting the data is quite narrow as compared to sets B and C. We can also deduce from the value of C that the learning machine is allowing many classification errors to exist.

Assigning the penalty to the misclassification given by C is a crucial step in SVM. A small value of C allows many misclassification errors and a large value of C allows minimal misclassification errors, thus allowing the learning machine to attain maximum accuracy. Figures 4.10 and 4.13 show the accuracies on data sets C and A.

In [87], H.T. Lin and C. J. Lin found that generally, the sigmoid kernel does not perform better than the RBF kernel. Table 4.2 compares the best accuracies using the sigmoid and RBF kernels. It can be seen that the sigmoid kernel matches the results from the RBF kernel with data sets B and C. In the case of set A, it performs reasonably well but not better than the RBF kernel. These results are in line with [87].

Table 4.2: Comparison of accuracies between sigmoid and gaussian kernels

Data	**Sigmoid Kernel**		**Gaussian Kernel**	
	best $(r, logC)$	best % accuracy	best $(\sigma, logC)$	best % accuracy
Set A	(-40,0.2)	72.22	(1,8)	85.55
Set B	(-2.8,10)	98.88	(1,8)	98.88
Set C	(-2.8,10)	100	(1,8)	100

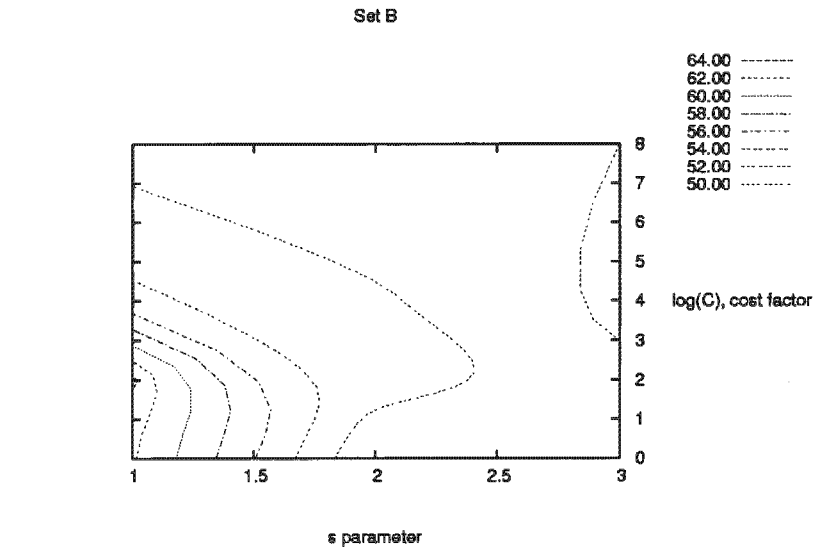

Figure 4.6: Sigmoidal grid search for set B

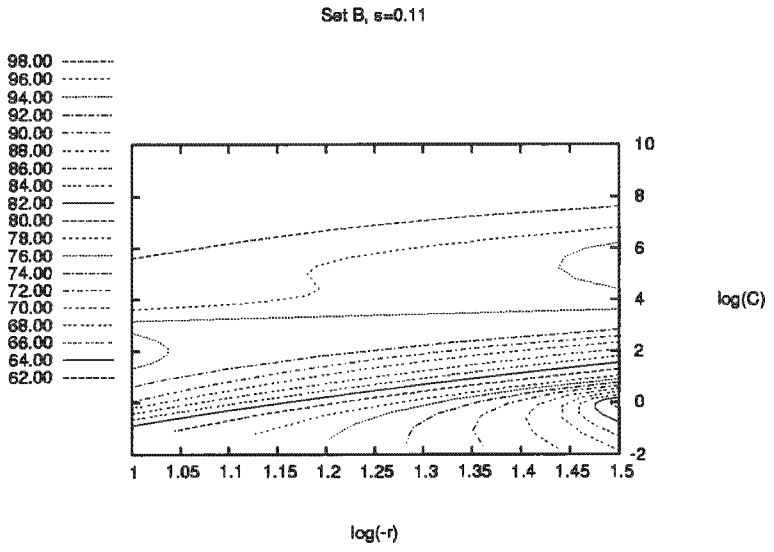

Figure 4.7: Refined grid search for set B

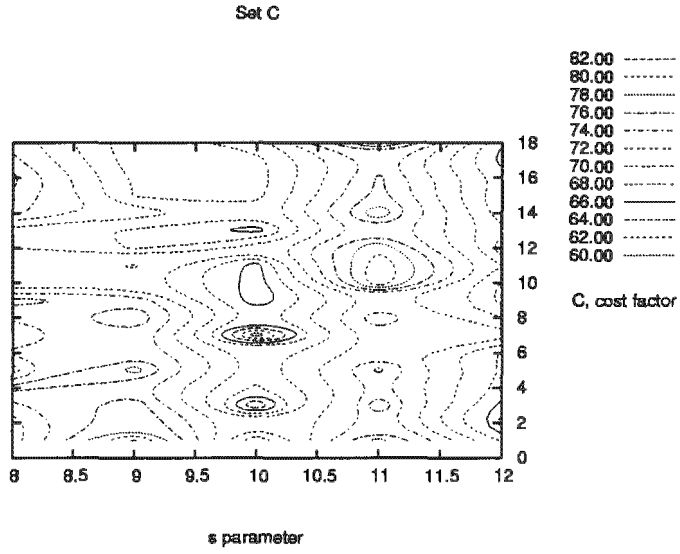

Figure 4.8: Coarse sigmoidal grid search on s and C for set C

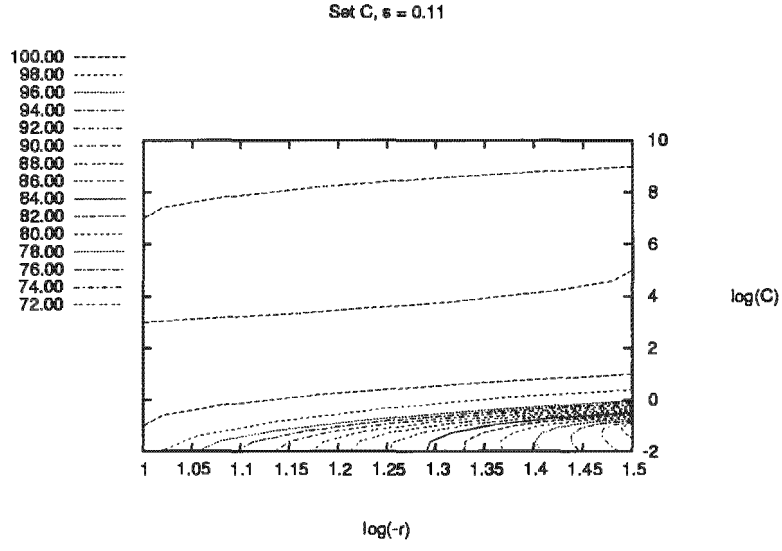

Figure 4.9: Refined sigmoidal grid search for set C

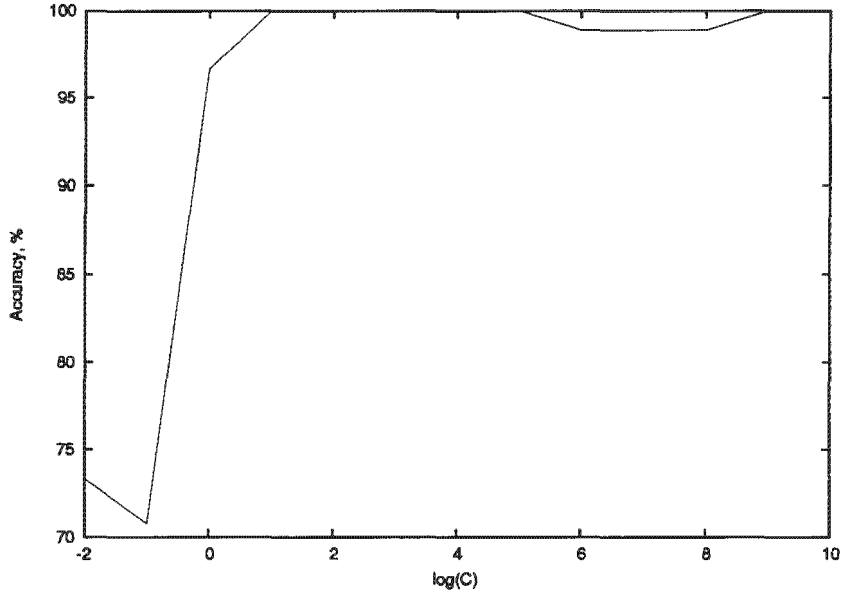

Figure 4.10: Accuracy with varying C for set C

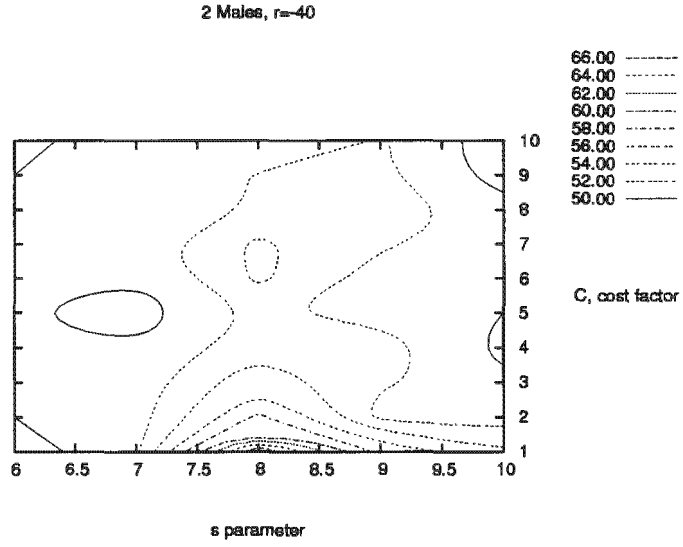

Figure 4.11: Sigmoidal grid search for set A

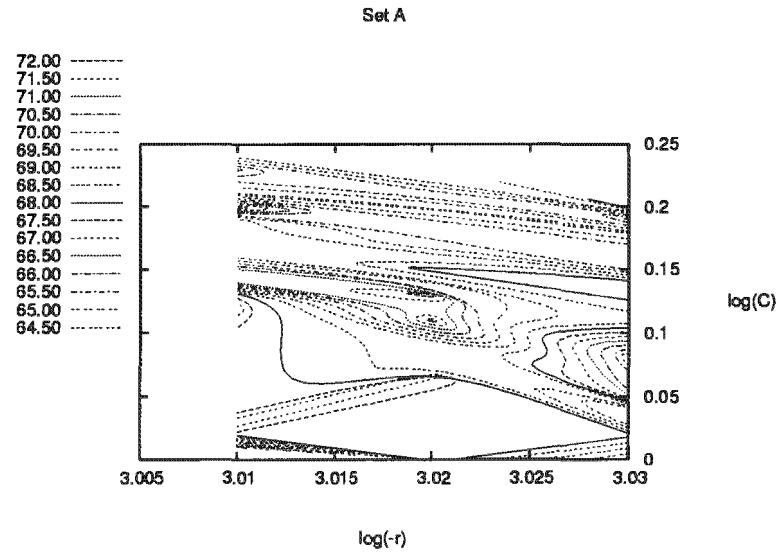

Figure 4.12: Refined grid search for set A

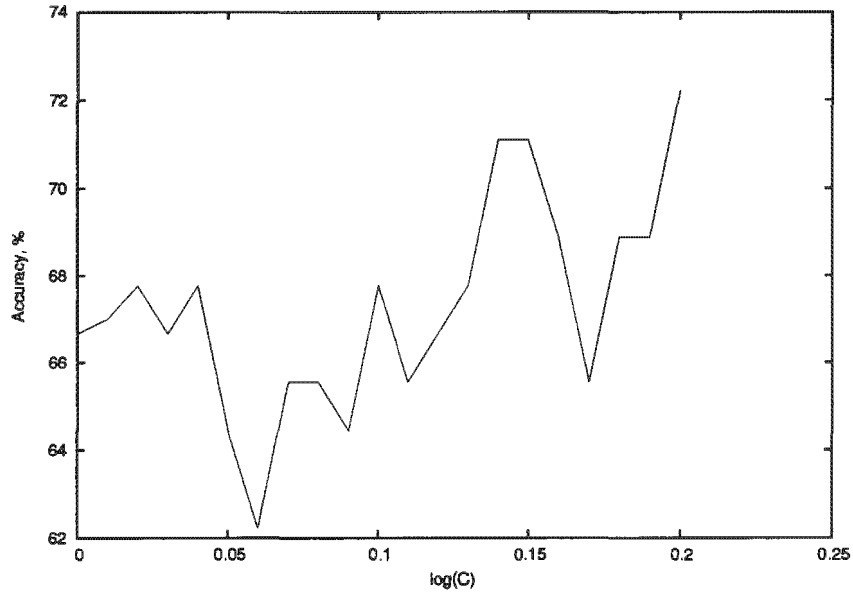

Figure 4.13: Accuracy with varying C for set A

4.3.1.3 Using a polynomial kernel

The polynomial kernel is given by $(s\,x.y + r)^d$. There are four parameters of interest in tuning the polynomial kernel. They are s, r, d, the degree of the polynomial and c. The grid search for this kernel is more computer extensive as there are four unknown parameters. In [143], the authors found that the polynomial kernel can lead to an optimization problem that has an ill-conditioned Hessian. They attribute this fact to the large order of the polynomial and to very high dimensional input data, thus rendering the value of the kernel excessively large. Table 4.3 shows the best accuracies for each set. It can be noted from the results that the best performance occurs at order 4 for each set. The same deduction can be made as before; the system confuses between the identity of male speakers to a great extent as compared to the other two sets.

Table 4.3: Accuracies using polynomial kernel

Data Set	**best** (r, s, C, d)	**best % accuracy**
Set A (males)	(-4,5,10,4)	72.22
Set B (females)	(-4,8,10,4)	98.88
Set C (male/female)	(-4,5,10,4)	100

Figures 4.14 and 4.15 show a 3-dimensional plot of the accuracy rates of set A data for orders 2 and 4 respectively. Figure 4.16 shows the accuracy of the learning machine as the order of the polynomial increases. It can be seen that the highest accuracy is obtained at degree 4. As the degree increases beyond 4, the polynomial kernel fails to classify the speakers. This is in accordance with [144].

An exhaustive search for the best parameters was performed for sets B and C and it was found that a polynomial of order 4 gave the best accuracy. The best accuracies are shown in table 4.3. Figures 4.17 and 4.18 show the accuracies of sets B and C respectively.

4.3.1.4 Using a linear kernel

The form of the linear kernel is $(x.y + 1)$. This kernel is the simplest of all the kernels considered in this study. The linear kernel has only one parameter, C. Since the data under investigation are not linearly separable, the linear kernel fails to classify the data. As C varies from 1 to 3000, the performance remains at 67.78% for set A, 33.33% for set B and 60.00% for set C. This shows that $C = 1$ is the finite limiting value in line with [73]. The linear kernel performs poorly compared to the other kernels because the separating plane between the two classes for each data set is not linear.

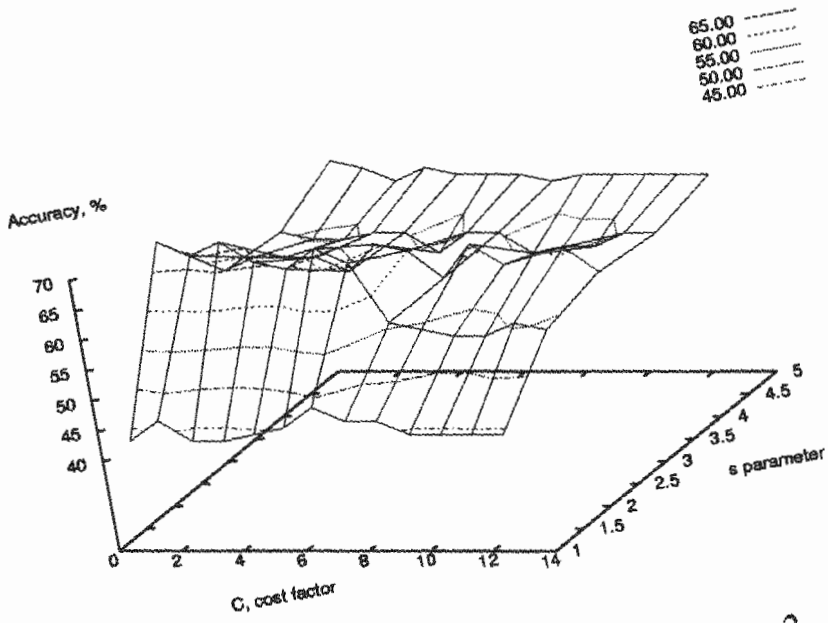

Figure 4.14: Performance for set A, r = -4, d = 2

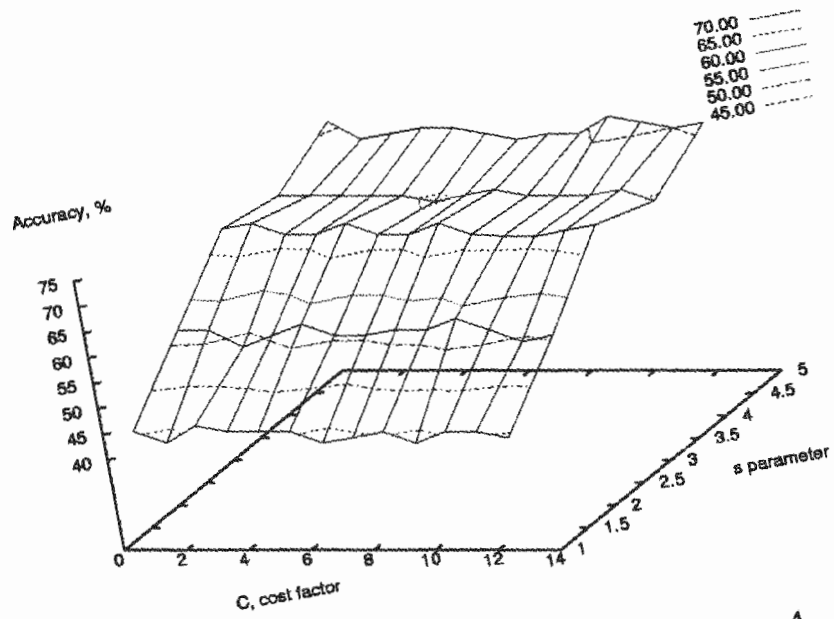

Figure 4.15: Performance for set A, r = -4, d = 4

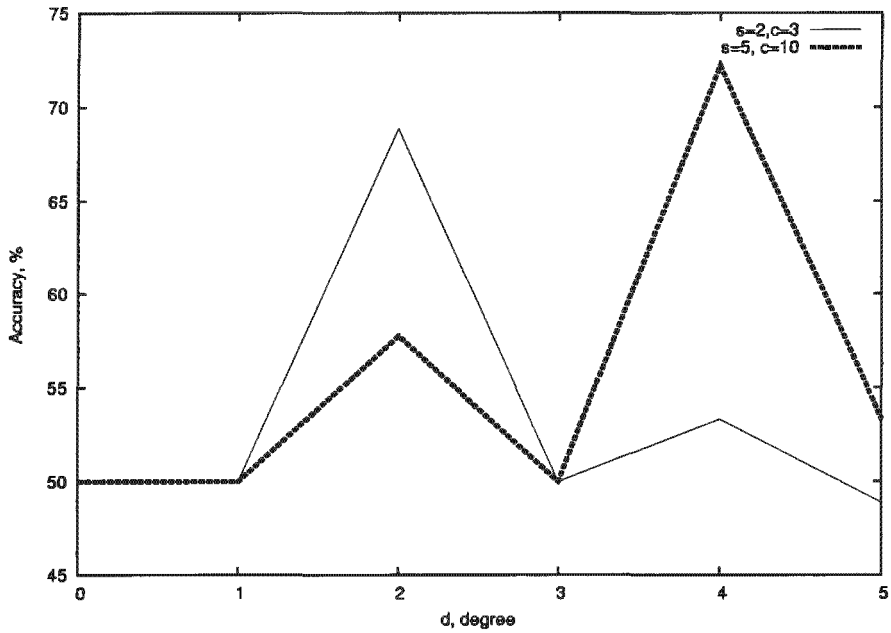

Figure 4.16: Accuracy for varying order of polynomial for set A

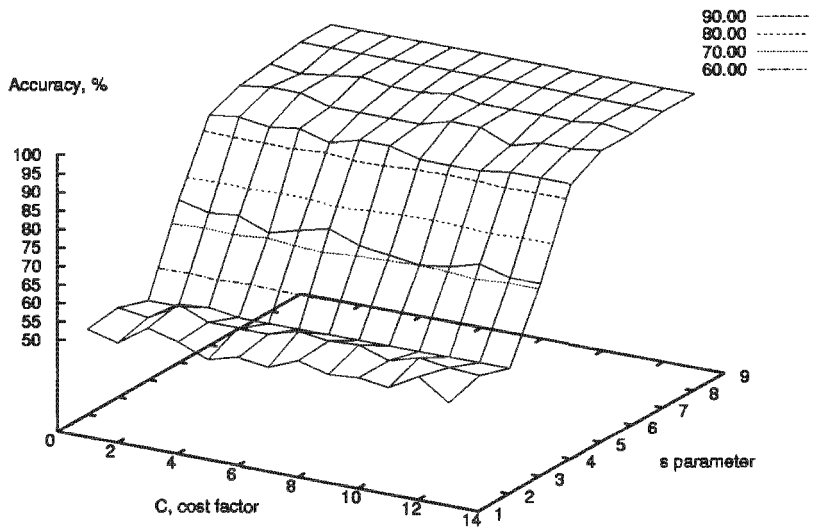

Figure 4.17: Performance for set B, r = -4, d = 4

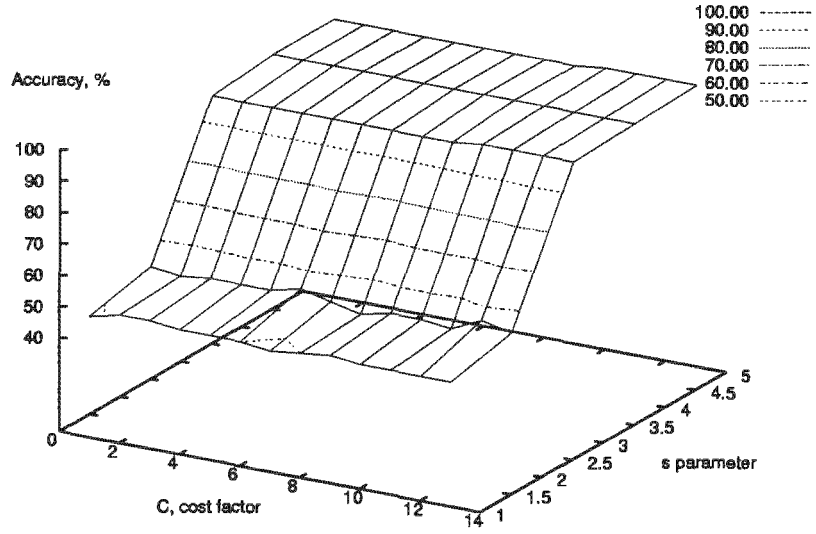

Figure 4.18: Performance for set C, r = -4, d = 4

4.3.2 Computational time

The experiments were performed on a 3.0GHz Pentium IV system. The execution time for each kernel is given in table 4.4. The times given are in minutes. The first observation that can be made is that the linear kernel took excessive time to converge. This is because of the same factor discussed previously. Besides the linear kernel, the gaussian kernel takes the longest amount of time to classify the data. The sigmoid kernel, however, is the least expensive in terms of computation time. It would seem advantageous to use the sigmoid kernel for a classification problem. However, the sigmoid kernel is non-PSD but CPD for some parameters only. Hence, the problem of finding the optimal hyperparameters becomes very computer intensive. The fact that gaussian kernels take longer to converge than other kernels is reported in [52]. The same observation was made in this study.

Table 4.4: Computational time (in minutes) of the different kernels

Kernels	Set A	Set B	Set C
Gaussian	60	56	35
Sigmoid	10	21	25
Polynomial	25	35	15
Linear	120	70	50

4.3.3 Discussion of results

From the results obtained, it can be observed that the linear kernel performs poorly. This can be attributed to the fact that the decision plane in the feature space separating the two classes is non-linear. Figure 4.19 shows a comparison of the various kernels. The bargraph does not include the linear kernel as it does not warrant any comparison due to its poor performance. It can be observed from the figure that for sets B and C the three kernels namely, gaussian, polynomial and sigmoid, have the same performance. For set A, however, the highest accuracy is obtained from the gaussian kernel. There is 13.33% difference between the gaussian kernel and the other two kernels for set A. Set A which includes the male speakers could not achieve performance as high as the other two sets.

The system could not clearly distinguish the male speakers. The gaussian kernel achieved an identification rate of 85.55% while the polynomial and sigmoid kernels achieved an accuracy of only 72.22%. One of the possible reasons for the "poor" performance on set A could be associated with the fact that the two males chosen in our experiments sound very much alike. Another cause could be that there is considerable broadband noise in the speakers' utterances. When two other male speakers with relatively less noise were used, an accuracy of 100% was obtained.

It can also be observed from the results of each data set that the optimal hyperparameters for each kernel are approximately the same. For the gaussian kernel, the parameters were the same irrespective of the data set. The optimal parameters for the polynomial kernel were also closely related in the feature space. The sigmoid kernel presented same parameters for sets B and C but set A's parameters were very far from the others. A shift of $r = -40$ on the x-axis is far from $r = -2.8$. Also, the low value of the penalty term ($C = 1.15$) indicates that the sigmoid kernel function allowed many misclassifications.

In [32], it is reported that ultimately each valid SVM kernel should have the same performance. It can be deduced that after an exhaustive search of optimal hyperparameters was performed for each kernel, approximately the same results were obtained on each data set.

As expected, the linear kernel took the longest amount of time to attempt to classify the data. The computational time of the gaussian kernel was also high when compared to the sigmoid and polynomial kernels as reported in [52]. The sigmoid kernel, being the least inexpensive in terms of training time, does not warrant its use for a classification task. Finding its optimal hyperparameters is very computationally expensive. The author would advise the use of the gaussian or the polynomial kernels. However, in the case of

the polynomial kernel, there are four tunable hyperparameters compared to only two for the gaussian kernel. Hence, the gaussian kernel presents a better alternative even though it takes quite long to compute.

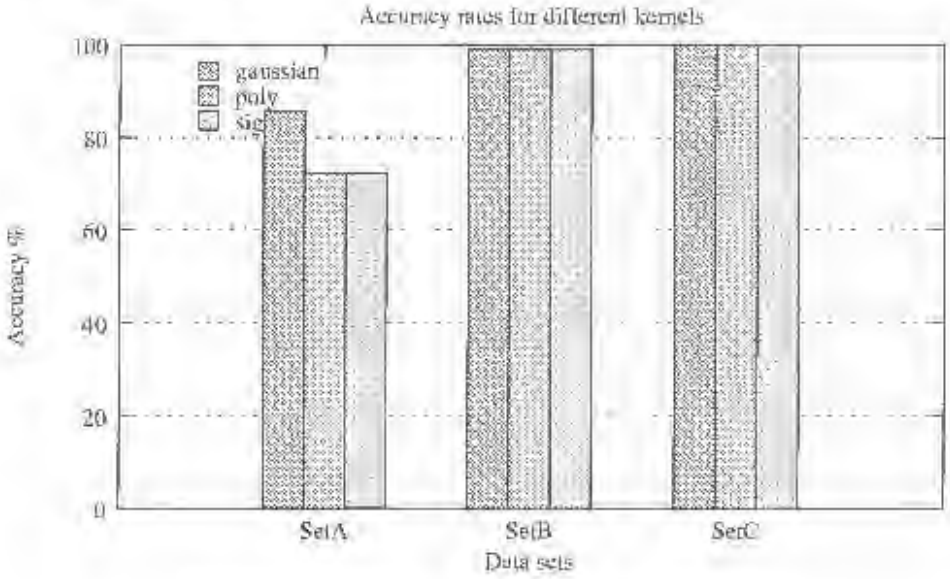

Figure 4.19: Performance of different kernels

4.4 Summary

This chapter has reported the experiments carried out in this study and the corresponding results. An exhaustive search for the best hyperparameters was performed and the results show that all the kernels, besides the linear kernel, achieve the same accuracy for each data set. Since the decision boundary of the two classes is not linear, the linear kernel performs poorly. Two types of scaling were used for the gaussian kernel and there was an increase in accuracy. The results also show that the gaussian kernel took the most amount of time to classify the data, after the linear kernel. The sigmoid kernel took the least time to converge, but much time was spent on finding the optimal parameters. Hence, the polynomial presents the best alternative based on the computational time taken to converge. However, since the gaussian kernel possesses the least number of tunable hyperparameters as compared to the sigmoid or polynomial kernels, it is advisable to use a gaussian kernel. Brown et al. [12] reported that "this is not conclusive evidence that the radial basis SVM is superior to other methods, but it is suggestive." . Table 4.5 shows a list of some previous research done on kernel selection. It can be seen that the gaussian and polynomial kernels are the two kernels that offer the best accuracy similar to the findings of this study.

Table 4.5: Previous research on kernel selection

Previous work	**RBF**	**Polynomial**	**Sigmoid**	**Linear**	**Best kernel**
Phoneme Classification [126]	√	√		√	RBF
Classification of gene data [128]	√	√	√	√	RBF
Prediction of protein [62]	√	√	√		RBF
Vowel Classification [52]	√	√			RBF
Handwritten digit recognition [11]	√	√			Polynomial
Speaker id/verification [143]	√	√			Polynomial
Digit image recognition [6]	√	√			Polynomial
Identification of top quark events [138]	√		√		Same
Classification of emotion [40]	√	√	√	√	Sigmoid

Chapter 5

Hybrid GMM-SVM speaker identification system

This chapter looks into a hybrid GMM-SVM on a speaker identification problem. It starts by giving an overview of gaussian mixture models (GMM). It then considers experiments done on the hybrid system and a discussion of the results obtained is presented.

5.1 Introduction

The GMM-SVM system uses the generative strength of GMM coupled with the discriminative power of SVM to improve the performance of a system. Fine et al [46] proposed such a system on a text-independent speaker identification task. Their baseline system, GMM, produced an N-best list based on GMM likelihood scores while the classification algorithm, SVM, uses that N-best list to output the speaker with the maximum score. They reported a 25% reduction in identification error rate (compared to the original GMM system) in using 52 speakers from the LLHDB database [124].

Fine et al also investigated similar hybrid systems in speaker verification tasks [47] and in digit recognition in a noisy environment [48]. Le and Bengio [81] also considered such a hybrid system on a speaker verification task. In all these cases, the hybrid approach outperforms the original GMM system. Mashao [95] presented N-best hybrid system on a text-independent speaker identification system. He reported a maximum performance of 74.6% in using 630 speakers from the NTIMIT [108] database for the N-best hybrid system.

Since SVM performs better with limited data [69, 97], it seems advantageous to use SVM to classify the top two speakers. Since the errors made by GMM and SVM are uncorrelated [47, 46, 48], this provides an attractive opportunity to complement the two systems. This chapter entails using the same principle in [95] for the hybrid system. The four SVM kernels dealt with in chapter 4 are applied in turn to the SVM machine. The optimum hyperparameters obtained in chapter 4 for each of the kernels are used in the hybrid system.

5.2 Brief description of GMM

Gaussian mixture models (GMM) represent a state-of-the-art technique for speaker identification tasks. The feature vectors of a speaker, obtained from the feature extractor PFS, are modeled by a GMM density [119]. Given a D-dimensional feature vector denoted as x, the mixture density is defined as:

$$p(x \mid \lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(x) \tag{5.1}$$

The Gaussian mixture density is a weighted sum of M component densities, $b_i^s(x)$ given by equation 5.2:

$$b_i^s(x) = \frac{1}{(2\pi)^{D/2} \mid \sum_i^s \mid^{1/2}} \times exp\{-\frac{1}{2}(x - \mu_i^s)(\sum_i^s)^{-1}(x - \mu_i^s)\} \tag{5.2}$$

The parameters of each density are the mean vector, μ_i^s, and diagonal covariance matrix, $\sum_i^s$. The mixture weights satisfy the constraint $\sum_{i=1}^{M} p_i = 1$. Each speaker is represented by a model, λ, where

$\lambda = \{p_i, \overrightarrow{\mu_i}, \sum_i\}$, $i = 1, ..., M$

The values of $M = 32$ and $D = 30$ are used in the experiments. Given the feature vectors from a speaker's training speech, the aim of the model training is to learn the parameters of the GMM. This is achieved by expectation-maximization (EM) algorithm. EM is an iterative algorithm which produces maximum likelihood (ML) estimates [37]. For a given set of speakers $S = \{1, 2, ..., s\}$ represented by models $\lambda_1, \lambda_2, ..., \lambda_s$, the goal is to find the speaker model which has the maximum a posteriori probability for a given observation sequence. A test vector is identified with speaker s if

$$\hat{s} = arg_{1 \leq s \leq S} \sum_{t=1}^{T} log(p(x_t \mid \lambda_s)) \tag{5.3}$$

5.3 Complementary GMM/SVM system

Eight utterances were used to train the GMM classifier and the remaining two utterances were used for testing. The identification of the top two speakers is performed by GMM. SVM, then, generates a model for each of the two speakers (generated from GMM) using data that were used for GMM modelling. The two test utterances are then matched to the models generated from SVM to identify the correct speaker. Mashao [95] reported a performance of 77.6% when the correct speaker is among the top two speakers using a GMM classifier.

Experiments were conducted by using SVM to complement GMM. The system was evaluated on 630 speakers from the NTIMIT database. The hybrid system performed a classification of only the top two speakers using the various kernels from chapter 4. A GMM identification rate of 70.2% is always obtained [95].

The optimization used in the experiments is the *confidence measure* of the classifier decision [95]. The confidence is given by d_0, where $d_0 = L_0 - L_1$ where L_0 and L_1 are the log-likelihood scores of the first and second speakers. If d_0 is very large, then the two speakers feature vectors are far apart in the feature space and the system is very confident that person 0 is the correct speaker.

5.4 Results of hybrid system

Figures 5.1, 5.3, 5.4 and 5.5 show the performance of the system using the gaussian, sigmoid, polynomial and linear kernels respectively. The linear kernel performed very poorly as expected. Figure 5.2 shows the performance of the hybrid system when scaling defined by equation 4.1 was used. A 1.4% increase in identification is observed when using scaling of equation 4.2.

Fine et al. [48] showed that the hybrid system works only if the second part of the hybrid system (SVM) gives higher performance than the baseline. From table 5.1, we can see that the gaussian kernel offers the best performance. A GMM identification rate of 70.2% is always obtained. GMM has a 'perplexity' of 630 while SVM has a 'perplexity' of only

Table 5.1: Performance of hybrid system using different SVM kernels

Kernels	Performance, %	Confidence value
Gaussian	72.5	63-69, 90, 91
Polynomial	71.1	38-40, 43-45, 68, 69, 73-77, 90, 91
Sigmoid	71.0	43-45, 68, 69

2. SVM is forced to make a decision based on the outputs from GMM. GMM produces the log likelihood scores of the two top speakers and SVM decides who is the unknown speaker amongst the two provided from GMM. Hence, if the unknown speaker is not amongst the top 2 speakers, SVM will still make a decision. This affects the performance of the SVM machine which in turn affects the overall performance of the hybrid system. The improvements in performance of the hybrid system, as compared to using GMM alone, are not so significant. A further reduction identification error rate of 7.7% was observed.

Table 5.2 shows the computation times (in hours) of the hybrid system when the SVM kernels were varied. The GMM/SVM hybrid system takes very long to identify the speakers. The hybrid system using the gaussian kernel SVM is the most computer-intensive. However, the highest performance (72.5%) is obtained from using that kernel.

Table 5.2: Comparison of computation time for hybrid GMM/SVM system

SVM Kernels	Time (hours)
Gaussian	7.5
Polynomial	5.17
Sigmoid	5.23
Linear	4.28

5.5 Summary

This chapter dealt with a hybrid GMM-SVM system. The GMM system generated the top two speakers and SVM was used to identify the unknown speaker from the top two speakers. A brief description of GMM was provided. Furthermore experiments on the hybrid system using the kernels analyzed in chapter 3 were performed and the results reported accordingly.

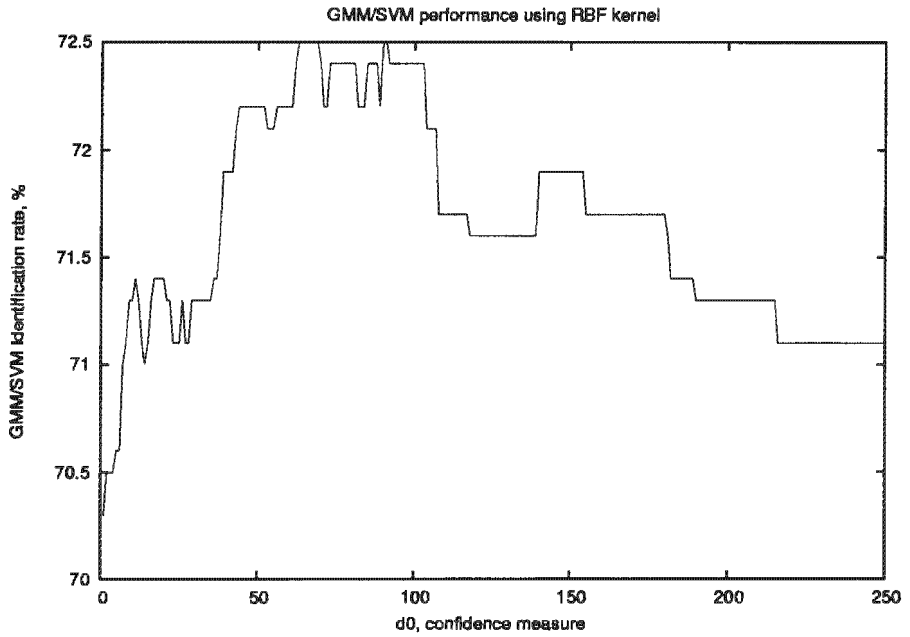

Figure 5.1: Performance as a function of confidence measure for top two speakers using scaling as shown in equation 4.2

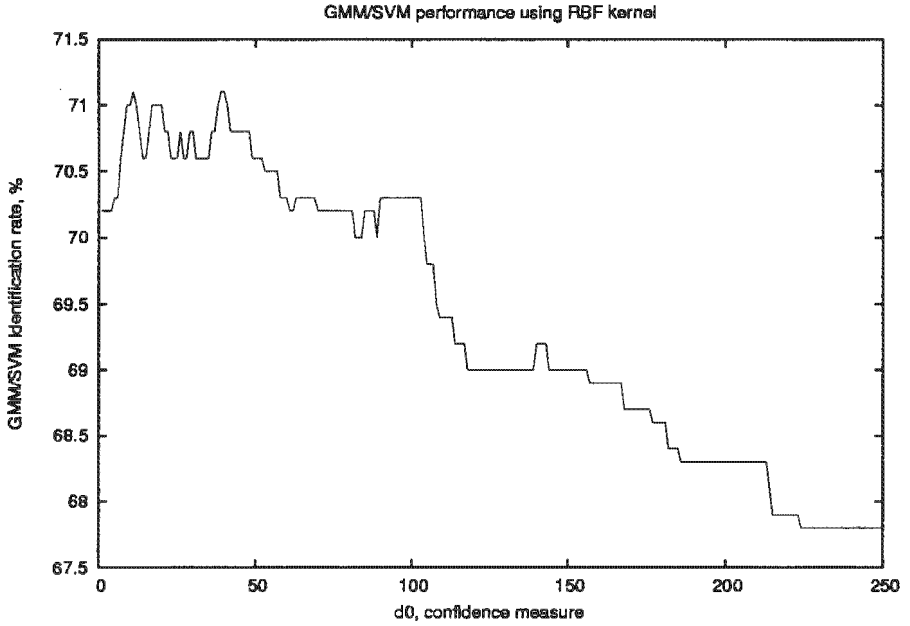

Figure 5.2: Performance as a function of confidence measure for top two speakers using scaling as shown in equation 4.1

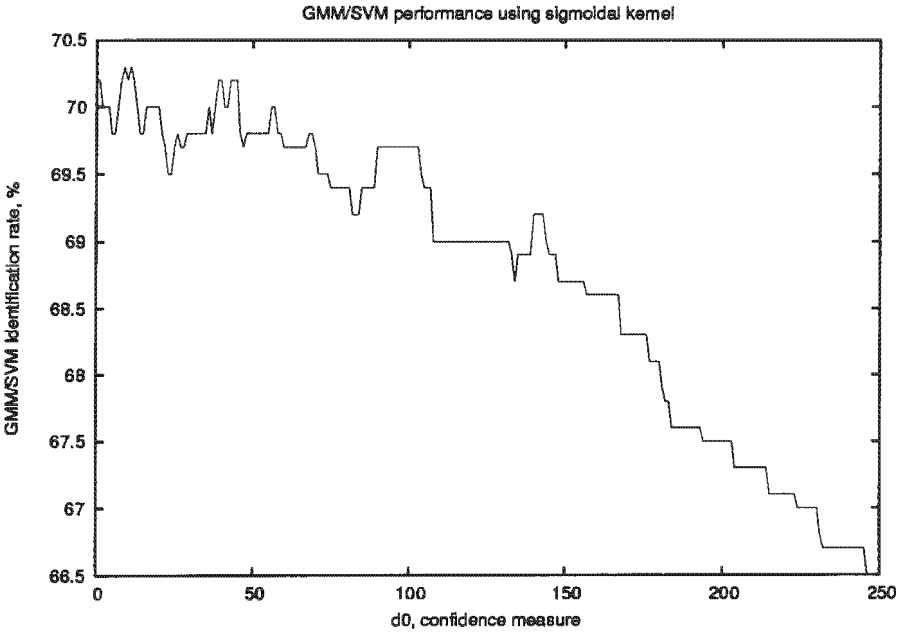

Figure 5.3: Performance as a function of confidence measure for top 2 speakers

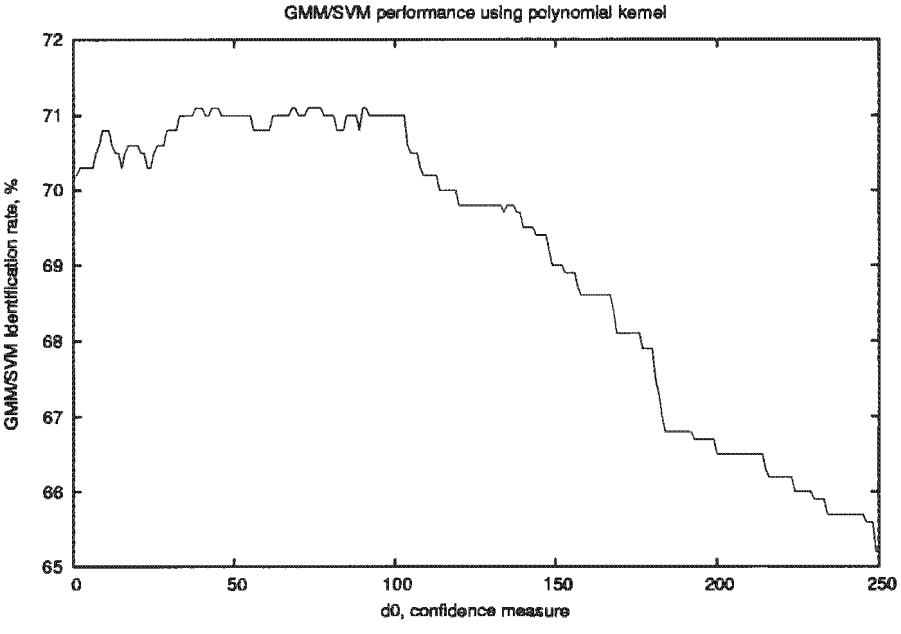

Figure 5.4: Performance as a function of confidence measure for top 2 speakers

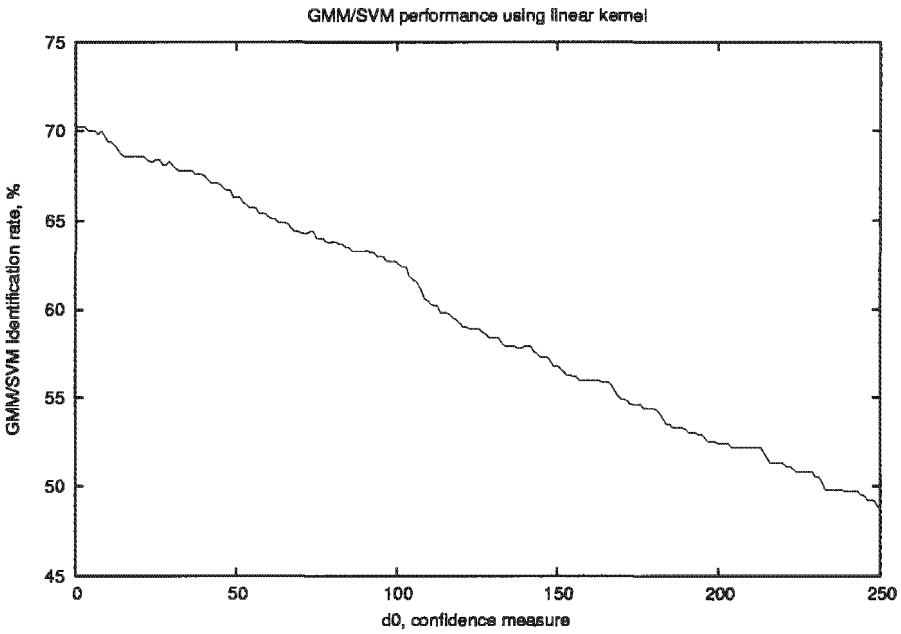

Figure 5.5: Performance as a function of confidence measure for top two speakers

Chapter 6

Conclusions

The goal of this thesis was to investigate four SVM kernels. Chapter 2 gave an overview of speech technology. Chapter 3 dealt with SVM which is the core of this thesis. In chapter 4, the experiments performed were described in detail. Chapter 5 showed the application of SVM to a hybrid system.

6.1 Comparison of kernels

We can conclude from the experiments performed that all the SVM kernels, besides the linear kernel, achieved approximately the same performance for each data set. This observation is in accordance with [32] where it is reported that ultimately each valid kernel should have the same performance. The type of kernel determines the data distribution in the feature space. Set A, however, was problematic as the speakers might possess considerable noise in their utterances. The linear kernel performed poorly due to the fact that the separating hyperplane differentiating the two classes is non-linear. Hence, there are many misclassified data.

Optimizing the performance of SVM algorithm entails the optimization of its parameters. We have shown that the SVM learning machine is dependent on the hyperparameters for each kernel. We have also shown that for each kernel, besides the sigmoid kernel, the hyperparameters were found to be approximately the same. Set A for the sigmoid kernel was problematic. This might be due to the heavy noise factor mentioned before and as such the sigmoid kernel, being CPD for only certain parameter values, could not consistently identify the male speakers.

A comparison of the computational time was also performed. It was shown that the linear kernel was the most computationally expensive followed by the gaussian kernel. The result for the linear kernel was expected. The result for the sigmoid kernel, though, was the least expected. The fact that the sigmoid kernel takes the least amount of time to classify the data does not necessitate its use. This is due to the fact that a considerable amount of time is spent trying to find the optimal hyperparameters as the sigmoid kernel is CPD for only certain hyperparameter values. The polynomial or gaussian could be used for any classification task but since the gaussian kernel has the least number of tunable hyperparameters, it is advisable to start with this kernel first. The fact that the gaussian kernel takes long to converge is also reported in [52]. Brown et al. [12] reported that "this is not conclusive evidence that the radial basis SVM is superior to other methods, but it is suggestive." .

6.2 Hybrid GMM/SVM system

The optimized hyperparameters obtained were then used in the implementation of a GMM/ SVM hybrid system. High performance was expected based on the accuracies obtained in chapter 4. However, the best performance achieved was only 72.5% using the gaussian kernel. This might be due to the fact that since we are using a hybrid system, a "pure" performance of SVM is not obtained. The latter is forced to make a decision based on the speakers provided from GMM. Even if the unknown speaker is not amongst the top two speakers, SVM will provide a decision. Hence, this affects the performance.

However, an improvement in performance was obtained with the hybrid system as compared to using GMM alone. The GMM classifier had an identification error rate of 29.8% with a "perplexity" of 630 while SVM possesses only a "perplexity" of 2. The identification error rate is reduced by 7.7%. This result is obtained when the gaussian kernel is used for the SVM machine. Hence, better performance is obtained using the gaussian SVM kernel at the expense of computational cost.

6.3 Recommendations

The following recommendations are made:

- Due to time constraint, the polynomial and sigmoid kernels could not be evaluated

by using the scaling described in equation 4.2. Therefore it is recommended that further research should use this type of scaling.

- Further study should use a normalized polynomial kernel to obtain optimal hyper-parameters. The procedure used in [142] should be used.
- This study evaluated the different SVM kernels using PFS as the front-end. Future work should try different front-ends with SVM as the back-end to evaluate the SVM kernels.
- A practical difficulty of any classification task is in the selection of the hyperparameters that embed a kernel function. The SVM machine takes excessive time to compute classification of data, hence an automatic model selection of kernel as described in [84] should be used where the parameter and model with the best loo rates are obtained .
- The SVM kernels described in this study did not take into account the distributions of the data. Hence, a new approach using probabilistic distance kernels, as described in [103] should be investigated.

Bibliography

[1] W. H. Abdulla and N. K. Kasabov. Improving speech recognition performance through gender separation. *Artificial Neural Networks and Expert Systems International Conference (ANNES)*, pages 218–222, 2001.

[2] E. Aizermann, E. Braverman, and L. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[3] A. Ali and A. Abraham. Improved kernel learning using smoothing parameter based linear kernel. In *IWANN*, pages 206–213, 2003.

[4] L. M. Arslan and J. H. L. Hansen. Language accent classification in american english. In *Speech Communications*, volume 18, pages 353–367, 1996.

[5] S. Avalone. http://www.netbytel.com/literature/e-gram/technical3.htm.

[6] N. Ayat, M. Cheriet, and C. Suen. Empirical error based optimization of svm kernels: Application to digit image recognition. In *8th IWFHR*, 2002.

[7] N. T. Baloyi. Comparison of features for large population speaker identification. Master's thesis, University of Cape Town, 2000.

[8] D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York, 1982.

[9] L. Besacier, J. F. Bonastre, and C. Fredouille. Localization and selection of speaker specific information with statistical modeling. In *Speech Communication*, pages 89–106, 2000.

[10] BioID. www.bioid.com.

[11] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[12] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares, and D. Haussler. Support vector machine classification of microarray gene expression data. Technical report, University of California, Santa Cruz, 1999.

[13] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. In *Proceedings of the National Academy of Sciences*, volume 97, pages 262–267, 2000.

[14] K. Brudzewski, S. Osowski, and T. Markiewicz. Classification of milk by means of an electronic nose and svm neural network. *Sensor and Actuators B*, 98:291–298, 2004.

[15] C. Burges. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*, volume 2, pages 121–167, 1998.

[16] C. Campbell. *An introduction to kernel methods*, chapter 7, pages 155–192. Springer Verlag, 2000.

[17] J. Campbell and D. A. Reynolds. Corpora for the evaluation of speaker recognition systems. In *Proceedings of IEEE ICASSP*, pages 2247–2250, 1999.

[18] J. P. Campbell. Speaker recognition: A tutorial. In *Proceedings of the IEEE*, volume 85, pages 1437–1462, 1997.

[19] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek. Phonetic speaker recognition with support vector machines. http://books.nips.cc/papers/files/nips16/NIPS2003SP01.pdf, 2003.

[20] C. Chang and C. Lin. Libsvm: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[21] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. In *IEEE Transactions on Neural Networks*, volume 10, pages 1055–1064, 1999.

[22] O. Chapelle and V. Vapnik. Choosing multiple parameters for support vector machines. In *Machine Learning*, volume 46, pages 131–159, 2002.

[23] U. V. Chaudhari, J. Navratil, G. Ramaswamy, and S. Maes. Very large population text-independent speaker identification using transformation enhanced multi-grained models. In *Proceedings of IEEE ICASSP*, May 2001.

[24] K. Chin. Support vector machines applied to speech pattern classification. Master's thesis, University of Cambridge, 1998.

[25] P. Clarkson and P. Moreno. On the use of support vector machines for phonetic classification. In *Proceedings of ICASSP*, volume 2, pages 585–588, 1999.

[26] A. Cohen and V. Lapidus. Unsupervised text independent speaker classification. In *Proceedings of the International Conference on Signal Processing Application and Technology*, pages 1745–1799, 1996.

[27] R. Collobert and S. Bengio. Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, pages 143–160, 2001.

[28] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[29] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience, 1953.

[30] N. Cristianini and C. Campbell. Dynamically adapting kernels in support vector mahines. In *Neural Information Processing Systems*, pages 204–210, 1998.

[31] N. Cristianini and B. Scholkopf. Support vector machines and kernel methods : the new generation of learning machine. In *AI Magazine*, volume 23, pages 31–41, 2002.

[32] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[33] A. Davis. http://www.techonline.com/community/relatedcontent/20044.

[34] K. Davis, R. Biddulph, and S. Balashek. Automatic recognition of spoken digits. In *J. Acoust. Soc. Am.*, volume 24, pages 637–642, 1952.

[35] D. Decoste and B. Scholkopf. Training invariant support vector machines. In *Machine Learning*, volume 46, pages 161–190, 2002.

[36] J. Deller, J. Hansen, and J. Proakis. *Discrete-Time Processing of Speech Signals*. IEEE Press, 2000.

[37] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[38] W. Dhaes and X. Rodet. Discrete cepstrum coefficients as perceptual features. In *Proceedings of International Computer Music Conference*, 2003.

[39] K. Duan, S. Keerthi, and A. Poo. Evaluation of simple performance measures for tuning svm hyperparameters. In *Neurocomputing*, pages 41–59, 2003.

[40] M. Dumas. Emotional expression recognition using support vector machines. CSE 254: Seminar on learning algorithms, 2001.

[41] W. Ebel and J. Picone. Human speech recognition performance on the 1994 csr spoke 10 corpus. In *Proceedings of the Spoken Language Systems Technology Workshop*, pages 53–59, 1995.

[42] I. El-Naqa, Y. Yang, M. Wernick, N. Galatsanos, and R. Nishikawa. Support vector machine learning for detection of microcalcifications in mammograms. In *Proc. Int. Conf. on Image Processing*, pages 953–956, 2002.

[43] English. http://www.buzzin.net/english/glossary.htm.

[44] V. Faber. Clustering and the continuous k-means algorithm. In *Los Alamos Science Journal*, number 22, pages 138–144, 1994.

[45] N. Fakotakis, J. Sirigos, and G. Kokkinakis. High performance text-independent speaker recognition system based on voiced/unvoiced segmentation and multiple neural nets. In *Proceedings of Eurospeech*, 1999.

[46] S. Fine, J. Navratil, and R. Gopinath. A hybrid gmm/svm approach to speaker identification. In *Proceedings of IEEE ICASSP*, 2001.

[47] S. Fine, J. Navratil, and R. A. Gopinath. Enhancing gmm scores using svm "hints". In *Proceedings of Eurospeech*, 2001.

[48] S. Fine, G. Saon, and R. Gopinath. Digit recognition in noisy environments via a sequential gmm/svm system. In *Proceedings of ICASSP*, 2002.

[49] J. L. L. Floch, C. Montacie, and M. J. Caraty. Speaker recognition experiments on the ntimit database. In *Proceedings of Eurospeech*, pages 379–382, 1995.

[50] J. L. L. Floch, C. Montacie, and M. J. Caraty. Gmm and arvm cooperation and competition for text-independent speaker recognition on telephone speech. In *Proceedings of ICSLP*, volume 4, 1996.

[51] J. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, UA, 1996.

[52] A. Ganapathiraju. *Support Vector Machines for Speech Recognition.* PhD thesis, Mississippi State, Mississippi, 2002.

[53] T. Ganchev, A. Tsopanoglou, N. Fakotakis, and G. Kokkinakis. Probabilistic neural networks combined with gmms for speaker recognition over telephone channels. In *Proceedings of 14th International Conference on DSP*, volume II, pages 1081–1084, July 2002.

[54] H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, pages 18–32, 1994.

[55] J. Godfrey, D. Graff, and A. Martin. Public databases for speaker recognition and verification. In *ESCA Workshop on Automatic speaker recognition 1994*, pages 39–42. 1994.

[56] Y. Guermeur. Combining discriminant models with new multi-class svms. Technical report, LORIA Campus Scientifique, Vandaeuvre-les-Nancy France, 2000.

[57] I. Guyon, B. Boser, and V. Vapnik. Automatic capacity tuning of very large vc-dimension classifiers. In *Neural Information Processing Systems 5*, pages 147–155, 1993.

[58] T. Hazen and V. Zue. Automatic language identification using a segment-based approach. In *Proceedings of Eurospeech*, pages 1303–1306, 1993.

[59] H. Hermansky. Perceptual linear predictive (plp) of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[60] C. Hsu and C. Lin. A comparison on methods for multi-class support vector machines. Technical report, National Taiwan University, Taiwan, 2001.

[61] C. Hsu and C. C. C. Lin. A practical guide to support vector classification. 2003.

[62] S. Hua and Z. Sun. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. In *Journal of molecular Biology*, volume 308, pages 397–407, 2001.

[63] J. Huan, W. Wang, A. Washington, J. Prins, R. Shah, and A. Tropsha. Accurate classification of protein structural families using coherent sungraph analysis. In *Pacific Symposium on Biocomputing*, volume 9, pages 411–422, 2004.

[64] X. Huang, A. Acero, and H. Hon. *Spoken language processing*. Prentice Hall PTR, 2001.

[65] D. Hush and B. Horne. Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 1993.

[66] hyperdictionary. http://www.hyperdictionary.com/medical/speech.

[67] E. T. S. institute. www.etsi.org. GSM CODEC.

[68] O. Ivansiuc. Support vector machine classification of the carcinogenic activity of polycyclic aromatic hydrocarbons. *Internet Electronic Journal of Molecular Design*, 1:203–218, 2002.

[69] R. Jhumka and D. Mashao. Comparing svm and gmm on a speaker identification task. In *Proc. of 13th Annual Symposium of the PRASA*, 2002.

[70] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Springer, editor, *Proceedings of the European Conference on Machine Learning*, pages 137–142, 1998.

[71] W. Kao, K. Chung, C. Sun, and C. Lin. Decomposition methods for linear support vector machines. *Neural Computation*, 2004. To appear.

[72] E. Karpov. Real-time speaker identification. Master's thesis, University of Joensuu, 2003.

[73] S. Keerthi and C. Lin. Asymptotic behaviours of support vector machines with gaussian kernel. *Neural Computation*, 15(7):1667–1689, July 2003.

[74] T. Kinnunen. *Spectral Features for Automatic Text-Independent Speaker Recognition.* PhD thesis, University of Joensuu, Finland, 2003.

[75] T. Kinnunen and P. Franti. Speaker discriminative weighting method for vq-based speaker identification. In *Proc. 3rd International Conference on audio and video-based biometric person authentication (AVBPA)*, pages 150–156, 2001.

[76] R. L. Klevans and R. D. Rodman. *Voice Recognition.* Artech House, 1997.

[77] S. Knerr, L. Personnaz, and G. Dreyfus. *Single-layer learning revisited: a stepwise procedure for building and training a neural network.* Springer-Verlag, 1990.

[78] J. Koolwaaij. *Automatic speaker verification in telephony: a probabilistic approach.* PhD thesis, University of Nijmegen, Netherlands, 2000.

[79] U. Krebel. *Pairwise classification and support vector machines.* MIT Press, 1999.

[80] H. Kuhn and A. Tucker. Nonlinear programming. In U. of California Press, editor, *Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilities*, pages 481–492, 1951.

[81] Q. Le and S. Bengio. Client dependent gmm-svm models for speaker verification. In *International Conference on Artificial Neural Networks, ICANN/ICONIP 2003*, 2003.

[82] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simcard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*, pages 53–60, 1995.

[83] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simcard, and V. Vapnik. Learning algorithms for classification: A comparison on handwritten digit recognition. In *Neural Networks: The Statistical Mechanics Perspective*, pages 261–276, 1995.

[84] J. Lee and C. Lin. Automatic model selection for support vector machines. Technical report, National Taiwan University, 2000.

[85] L. Lerato. Hierarchical methods for large population speaker identification using telephone speech. Master's thesis, University of Cape Town, 2003.

[86] L. Lerato and D. J. Mashao. Hierarchical approach for improving speaker identification. In *Proc. of 13th Annual Symposium of the PRASA*, pages 51–55, 2002.

[87] H. Lin and C. Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, 2003.

[88] S. H. Maes. Conversational biometrics. In *Proceedings of Eurospeech*, 1999.

[89] I. Magrin-Chagnolleau, G. Gravier, M. Seck, O.Boeffard, R. Blouet, and F. Bimbot. A further investigation on speech features for speaker characterization. In *Proceedings of ICSLP*, 2000.

[90] J. D. Markel and A. H. Gray. *Linear Prediction of Speech.* Springer-Verlag, 1976.

[91] J. Marques and P. Moreno. A study of musical instrument classification using gaussian mixture models and support vector machines. Technical report, Cambridge Research Laboratory, 1999.

[92] T. Martin, A. Nelson, and H. Zadell. Speech recognition by feature abstraction techniques. Technical report, Air Force Avionics Lab, 1964.

[93] D. Mashao. *Computations and Evaluations of an Optimal Feature-set for an HMM-based Recognizer.* PhD thesis, PhD. Brown University, 1996.

[94] D. Mashao. A gmm-svm speaker identification system. 2003.

[95] D. Mashao. N-best hybrid gmm-svm speaker identification system. 2003.

[96] D. J. Mashao. Parallel processing for the auditory feature-set of a speaker recognition system. In *Proceedings of 2002 SCI-ISAS*, 2002.

[97] D. J. Mashao. Comparing svm and gmm on parametric feature-sets. In *Proceedings of the 14th Annual Symposium of the PRASA*, pages 15–20, 2003.

[98] D. J. Mashao and N. T. Baloyi. Improvements in the speaker identification rate using feature-sets on a large population database. In *EUROSPEECH Proceedings*, volume 4, pages 2833–2836, 2001.

[99] A. Mayoraz and E. Alpaydin. Support vector machine for multiclass classification. In *Proceedings of the International Workshop on Artificial Neural Networks (IWANN'99)*, 1999.

[100] S. Molau, M. Pitz, R. Schluter, and H. Ney. Computing mel-frequency cepstal coefficients on the power spectrum. In *Proceedings of the IEEE ICASSP*, pages 73–76, 2001.

[101] M. Momma, M. Song, and J. Bi. Mining microarray gene expression data. http://www.cs.rpi.edu/ bij2/doc/dmproj.pdf. 2001.

[102] E. Monte, J. Hernando, X. Miro, and A. Adolf. Text independent speaker identification on noisy environments by means of self organizing maps. In *Proceedings of ICSLP*, volume 3, 1996.

[103] P. Moreno and P. Ho. A new svm approach to speaker identification and verification using probabilistic distance kernels. Technical report, HP Laboratories Cambridge, 2004.

[104] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using support vector machines. In *Neural Network for Signal Processing VII: Proceedings of the IEEE Signal Processing Society Workshop*, pages 511–520, 1997.

[105] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. In *IEEE Transactions on Neural Networks*, volume 12, pages 181–202, 2001.

[106] K. Muller, J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the Seventh International Conference on Artificial Neural Networks*, pages 999–1004, 1997.

[107] H. A. Murthy, F. Beaufays, L. Heck, and M. Weintraub. Robust text-independent speaker identification over telephone channels. In *IEEE Transaction of Speech and Audio Processing*, volume 7, pages 554–568, 1999.

[108] NYNEX. http://www.ldc.upenn.edu/catalog/docs/ldc93s2/ntimit.txt.

[109] H. Olson and H. Belar. Phonetic typewriter. In *J. Acoust. Soc. Am.*, volume 28, pages 1072–1081, 1956.

[110] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997a.

[111] Z. Pan, K. Kotani, and T. Ohmi. A on-line hierarchical method of speaker identification for large population. In *IEEE Proceedings of Nordic Signal Processing Symposium*, pages 33–36, 2000.

[112] E. Parris and M. Carey. Language independent gender identification. In *IEEE ICASSP*, pages 685–688, 1996.

[113] T. Parsons. *Voice and Speech Processing*. McGraw-Hill, 1986.

[114] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993.

[115] V. Radova and J. Psutka. An approach to speaker identification using multiple classifiers. In *Proceedings of IEEE ICASSP*, pages 1135–1138, 1997.

[116] R. P. Ramachandran, K. R. Farrell, R. Ramachandran, and R. J. Mammone. Speaker recognition - general classifier approaches and data fusion methods. In *The Journal of the Pattern Recognition Society*, pages 2801–2821, 2002.

[117] D. Reddy. An approach to computer speech recognition by direct analysis of the speech wave. Technical report, Stanford University, Computer Science Dept., 1966.

[118] D. Reynolds. Automatic speaker recognition using gaussian mixture models. In *The Lincoln Laboratory Journal*, volume 8, pages 173–192, 1995.

[119] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.

[120] D. Reynolds. An overview of automatic speaker recognition technology. In *Proceedings of ICASSP, IEEE*, volume IV, pages 4072–4075, 2002.

[121] D. Reynolds and R. Rose. Robust text-independent speaker identification using gaussian mixture models. In *IEEE transactions on Speech and Audio Processing*, volume 3, pages 72–83, 1995.

[122] D. A. Reynolds. Effects of population size and telephone degradations on speaker identification performance. In *Proceedings of the SPIE Conference on Automatic Systems for the Identification and Inspection os Humans*, July 1994.

[123] D. A. Reynolds. Large population speaker identification using clean and telephone speech. In *IEEE Signal Processing Letters*, volume 2, pages 46–48, 1995.

[124] D. A. Reynolds. Htimit and llhdb: Speech corpora for the study of handset transducer effects. In *Proceedings of ICASSP*, pages 1535–1538, 1997.

[125] P. Rose. *Forensic Speaker Identification.* Taylor and Francis, 2002.

[126] J. Salomon. Support vector machines for phoneme classification. Master's thesis, University of Edinburgh, 2001.

[127] V. Sanchez. Advanced support vector machines. In *Neurocomputing*, volume 55, pages 5–20, 2003.

[128] E. Schabell. Svm and bio-informatics: A look at microarray gene expression data. February 2002.

[129] M. Schmidt. Identifying speakers with support vector networks. In *Proceedings of Interface*, 1996.

[130] M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *Proceedings of ICASSP, IEEE*, pages 105–108, 1996.

[131] M. Slaney. Baby ears: A recognition system for affective vocalizations. In *Proceedings of ICASSP*, 1998.

[132] J. Soong, F. Rosenberg, and L. Rabiner. A vector quantization approach to speaker recognition. In *Proceedings of ICASSP*, pages 387–390, 1985.

[133] R. A. B. Soria and E. F. Cabral. Combining neural networks paradigms and mel-frequency cepstral coefficients correlations in a speaker recognition task. In *Proceedings of the International Conference on Signal Processing Applications and Technology*, 1996.

[134] S. Srivastava. Fundamentals of linear prediction. The lecture: Mississippi State University Elec.Eng., 1999.

[135] C. Tanprasert, C. Wutiwiwatchai, and S. Sae-tang. Text-dependent speaker identification using neural networks on distinctive thai tone marks. In *NECTEC Technical Journal*, volume 1, pages 249–253, 2000.

[136] A. Toutios and K. G. Margaritis. Development of a text-dependent speaker identification system with the ogi toolkit. In *Proceedings of the Conference on AI, SETN-2002*, pages 525–530, 2002.

[137] J. Turian. Svms in practice. SCLT lecture: New York University Computer Science Dept., 1999.

[138] A. Vaiciulis. Support vector machines in analysis of top quark production. In *Proceedings of the VIII International Workshop on Advanced Computing and Analysis Techniques in Physics Research*, volume 502, pages 492–494, 2003.

[139] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[140] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[141] T. Vintsyuk. Speech discrimination by dynamic programming. In *Kibernetika*, volume 4, pages 81–88, 1968.

[142] V. Wan. *Speaker Verification using Support Vector Machines*. PhD thesis, University of Sheffield, United Kingdom, 2003.

[143] V. Wan and W. Campbell. Support vector machines for speaker verification and identification. In *Proceedings Neural Networks for Signal Processing X*, pages 775–784, 2000.

[144] V. Wan and S. Renals. Evaluation of different kernels for speaker verification and identification. In *Proceedings of ICASSP, IEEE*, pages 669–672, 2002.

[145] A. Watkins. Timbre. The lecture: University of Reading, Department of Psychology, 2002.

[146] F. A. Westall, R. D. Johnston, and A. V. Lewis. Speech technology for telecommunications. In *BT Technol Journal*, volume 14, pages 9–27, 2001.

[147] T. Weston. http://florin.stanford.edu/ t361/fall2000/tweston/history.html.

[148] wordiQ. http://www.wordiq.com/dictionary/speech.html.

[149] C. Wu and J. Chen. Text-independent speaker identification based on small training data and fast search algorithms. In *Journal of Information Science and Engineering*, volume 11, pages 73–87, 1995.