KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Automatic Speaker Characterization

Automatic Identification of Gender, Age, Language and Accent from Speech Signals

Mohamad Hasan Bahari

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

May 2014

Automatic Speaker Characterization

Automatic Identification of Gender, Age, Language and Accent from Speech Signals

Mohamad Hasan BAHARI

Examination committee:
Prof. dr. ir. Carlo Vandecasteele, chair
Prof. dr. ir. Hugo Van hamme, supervisor
Prof. dr. ir. Patrick Wambacq, co-supervisor
Prof. dr. ir. Dirk Van Compernolle
Prof. dr. ir. Marc Moonen
Prof. dr. David van Leeuwen
(Radboud University Nijmegen, The Netherlands)
Dr. Najim Dehak
(Massachusetts Institute of Technology, USA)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

May 2014

ISBN 978-94-6018-838-1

D/2014/7515/62

Preface

Listening to an utterance raises judgments about speaker characteristics such as gender, dialect, age and emotional state. However, the capability of making accurate judgments differs from person to person and requires expert knowledge. In this research, we tried to develop accurate methods to identify different characteristics of speakers automatically. Although this is an ambitious goal and requires dealing with many challenges, its wide range of commercial, forensic and medical applications motivated us to explore this field. This work was supported by the European Commission through the Marie-Curie ITN-project Bayesian Biometrics for Forensics (BBfor2). As a Marie-Curie fellow, I had the opportunities to pursue the project goals by visiting the Computer Science and Artificial Intelligence Laboratory of Massachusetts Institute of Technology (MIT) and the Centre for Language and Speech Technology, Radboud University Nijmegen (RUN).

This thesis could not be successful without the help and the support of many people. First of all, I would like to express my deep gratitude to my supervisor Prof. Hugo Van hamme and co-supervisor Prof. Patrick Wambacq for their continuous support and enthusiastic help. The knowledge, intuition and advice of Prof. Van hamme have inspired me to accomplish this long journey.

I would like to thank my co-supervisor within the BBfor2 network Prof. David van Leeuwen for his help, support and useful suggestions.

Many thanks to Dr. Mitchell Mclaren, who brought me to the amazing research area of the i-vector framework, provided me the required software and guided me during my visit to RUN.

I would as well give my gratitude to my supervisors at MIT, Dr. Najim Dehak and Prof. Jim Glass for supporting my visit, valuable advice and constant encouragement. Development and evaluation of non-negative factor analysis framework was not possible without the diligence and scrutiny of Dr. Najim Dehak and Prof. Van hamme.

Thanks to Dr. Kris Demuynck, Dr. Rahim Saeidi, Prof. Lucas Burget, Prof. Jim Glass, Dr. Douglas Reynolds, Dr. Fred Richardson and Dr. Pedro Torres-Carrasquillo for their help, useful discussions, insightful questions and valuable comments on my work.

Thanks to Prof. David van Leeuwen, Dr. Najim Dehak, Prof. Marc Moonen and Prof. Dirk Van Compernolle for their willingness to serve on my jury and for their comments and suggestions to improve the thesis.

I am grateful to all members of the SPRAAK group for building up a friendly working environment and interesting discussions.

Abstract

Speech signals carry important information about a speaker such as age, gender, language, accent and emotional/psychological state. Automatic recognition of speaker characteristics has a wide range of commercial, medical and forensic applications such as interactive voice response systems, service customization, natural human-machine interaction, recognizing the type of pathology of speakers, and directing the forensic investigation process. This research aims to develop accurate methods and tools to identify different physical characteristics of the speakers. Due to the lack of required databases, among all characteristics of speakers, our experiments cover gender recognition, age estimation, language recognition and accent/dialect identification. However, similar approaches and techniques can be applied to identify other characteristics such as emotional/psychological state.

For speaker characterization, we first convert variable-duration speech signals into fixed-dimensional vectors suitable for classification/regression algorithms. This is performed by fitting a probability density function to acoustic features extracted from the speech signals. Since the distribution of acoustic features is complex, Gaussian mixture models (GMM) are applied to model the distribution of acoustic features. Due to lack of data, it is not possible to build a separate acoustic model for short utterances. Therefore, parametric utterance adaptation methods have been applied to adapt the universal background model (UBM) to the characteristics of utterances. The parameters of each adapted GMM characterize the corresponding utterance. An effective approach involves adapting UBM to speech signals using the Maximum-A-Posteriori (MAP) scheme. Then, the Gaussian means of the adapted GMM are extracted and concatenated to form a Gaussian mean supervector for the given utterance. Finally, a classification or regression algorithm is used to identify the speaker characteristics. While effective, Gaussian mean supervectors are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. In the field of speaker recognition, recent advances using the i-vector framework have increased the classification

accuracy considerably. This framework, which provides a compact representation of an utterance in the form of a low-dimensional feature vector, applies a simple factor analysis on GMM means. Motivated by this success, the i-vector framework is applied to the age estimation problem. In this approach, each utterance is modeled by its corresponding i-vector. Then, a within-class covariance normalization (WCCN) technique is used for session variability compensation. Finally, a least squares support vector regression (LSSVR) is applied to estimate the age of speakers. The proposed method is trained and tested on telephone conversations of the National Institute for Standard and Technology (NIST) 2010 and 2008 speaker recognition evaluation databases. Evaluation results show that the proposed method yields significantly lower mean absolute estimation error and a higher Pearson correlation coefficient between chronological speaker age and the estimated speaker age comapred to different conventional schemes. Finally, the effect of some major factors influencing the proposed age estimation system, namely utterance length and spoken language are analyzed.

Our experiments on age estimation show that GMM weights carry important information about the speaker. However, the state-of-the-art language/speaker recognition systems usually do not use this information. In this research, a non-negative factor analysis (NFA) approach is developed for GMM weight decomposition and adaptation. This modeling suggests a new low-dimensional utterance representation method, which uses a factor analysis similar to that of the i-vector framework. The obtained subspace vectors are then applied in conjunction with i-vectors to the language/dialect recognition problem. The suggested approach is evaluated on the NIST 2011 and RATS language recognition evaluation (LRE) corpora and on the QCRI Arabic dialect recognition evaluation (DRE) corpus. The assessment results show that the proposed adaptation method yields more accurate recognition results compared to three conventional weight adaptation approaches, namely maximum likelihood re-estimation, non-negative matrix factorization, and a subspace multinomial model. Experimental results also show that the intermediate level fusion of i-vectors and NFA subspace vectors improves the performance of the state-of-the-art i-vector framework.

Motivated by the success of the NFA framework in Language/dialect recognition we introduce a hybrid architecture of the NFA approach and the i-vector frameworks for the speaker age estimation problem. Evaluation on the NIST 2010 and 2008 SRE corpora shows that the proposed hybrid architecture improves the results of the i-vector framework considerably.

Abbreviations

ABI	Accents of the British Isles
ASR	Automatic Speech Recognition
CGN	Corpus Gesproken Nederlands
CMD	Cumulative Probability Mass Distributions
DBN	Deep Belief Nets
DRE	Dialect Recognition Evaluation
fNAP	feature Nuisance Attribute Projection
GMM	Gaussian Mixture Models
GMS	Gaussian Mean Supervector
GPPS	Gaussian Posterior Probability Supervector
GRNN	General Regression Neural Networks
HMM	Hidden Markov Model
HWNN	Hybrid WSNMF and GRNN
ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
LRE	Language Recognition Evaluation
LSSVM	Least Squares Support Vector Machine
LSSVR	Least Squares Support Vector Regression
MAP	Maximum A Posteriori
MD	Probability Mass Distributions
ML	Maximum Likelihood
MFFC	Mel Frequency Cepstral Coefficients
MIDA	Mutual Information Discriminant Analysis
MITLL	MIT Lincoln Laboratory
MSA	Modern Standard Arabic
NBC	Naive Bayesian Classifier
NFA	Non-negative Factor Analysis
NIST	National Institute of Standards and Technology
NMF	Non-negative Matrix Factorization

OD	Ordinal Distance
PCA	Principal Component Analysis
PPRLM	Parallel PRLM
PRLM	Phone Recognizer followed by Language Models
QCRI	Qatar Computing Research Institute
QP	Quadratic Programming
RATS	Robust Automatic Transcription of Speech
RBF	Radial Basis Function
SBS	Special Broadcast Services
SDC	Shifted Delta Cepstral
SMM	Subspace Multinomial Model
SNMF	Supervised NMF
SRC	Sparse Representation Classifier
SRE	Speaker Recognition Evaluations
SVM	Support Vector Machines
SVR	Support Vector Regression
UBM	Universal Background Model
VTLN	Vocal Tract Length Normalization
WCCN	Within Class Covariance Normalization
WPPCA	Weighted-Pairwise PCA
WSNMF	Weighted Supervised NMF

Contents

Abstract iii

Abbreviations v

Contents vii

List of Figures xiii

List of Tables xv

1 Introduction 1
- 1.1 Motivations and Goals . . . 1
- 1.2 Ethical Issues . . . 2
- 1.3 Challenges . . . 3
- 1.4 Related Work . . . 3
- 1.5 Problem Formulation . . . 4
- 1.6 Model-Based Speaker Characterization . . . 5
  - 1.6.1 Statistical Modeling and Signal Representation . . . 5
- 1.7 Template-Based Speaker Characterization . . . 13
- 1.8 Summary of Contributions . . . 13
- 1.9 Outline of the Thesis . . . 15
- 1.10 References . . . 17

2 Age estimation from telephone speech using i-vectors 23
- 2.1 Abstract . . . 24
- 2.2 Introduction . . . 24
  - 2.2.1 Related Work . . . 25
  - 2.2.2 Motivations, Goals and Summary of Contributions . . . 26
- 2.3 Age Estimation from Speech . . . 27
  - 2.3.1 Baseline Approaches . . . 27
- 2.4 Age Estimation using i-vectors . . . 27

2.4.1 Regression . . . 28
2.4.2 The i-vector framework . . . 31
2.4.3 i-vector Session Compensation . . . 32
2.4.4 Train and Test . . . 32
2.5 Experimental Setup . . . 33
2.5.1 Database . . . 33
2.5.2 Performance Metric . . . 35
2.6 Results and Discussion . . . 35
2.6.1 SVR and LSSVR . . . 36
2.6.2 Baseline Systems Results . . . 37
2.6.3 i-vectors for Age Estimation . . . 38
2.6.4 The Effect of Utterance Length . . . 40
2.6.5 The Effect of Language . . . 40
2.7 Conclusions . . . 42
2.8 Appendix I . . . 43
2.9 References . . . 44

3 Gender and age recognition using Gaussian weights 49
3.1 Abstract . . . 50
3.2 Introduction . . . 50
3.3 Background . . . 51
3.3.1 Weighted Supervised NMF . . . 51
3.3.2 LSSVR . . . 54
3.4 Proposed Approach . . . 54
3.4.1 Feature selection, acoustic model and supervectors . . . 54
3.4.2 Training Phase . . . 55
3.4.3 Testing Phase . . . 56
3.5 Evaluation and Results . . . 57
3.5.1 Corpus . . . 57
3.5.2 Results . . . 57
3.6 Conclusions . . . 59
3.7 References . . . 59

4 Accent recognition using i-vectors, Gaussian weights and Gaussian means 63
4.1 Abstract . . . 64
4.2 Introduction . . . 64
4.3 Related Work and Contributions . . . 65
4.4 System Description . . . 66
4.4.1 Problem Formulation . . . 66
4.4.2 Utterance Modelling Approaches . . . 67
4.4.3 Classifiers . . . 68
4.4.4 Training and Testing . . . 69

4.5 Experimental Setup 69
4.5.1 Database 69
4.5.2 Performance Measure 71
4.6 Results 72
4.6.1 Feature Level Fusion 73
4.7 Conclusions 74
4.8 References 74

5 Language and dialect recognition using non-negative factor analysis 79
5.1 Abstract 80
5.2 Introduction 80
5.3 Background 82
5.3.1 Problem Formulation 82
5.3.2 Universal Background Model 83
5.3.3 i-vector Framework 83
5.3.4 Conventional GMM Weight Adaptation Approaches . . 84
5.4 Non-negative Factor Analysis 86
5.4.1 Updating Subspace Vector $\mathbf{r}$ 87
5.4.2 Updating Subspace Matrix $\mathbf{L}$ 90
5.5 Comparison between NMF, SMM and NFA 91
5.5.1 Modeling 91
5.5.2 Computation and Initialization 94
5.6 Experiments and Results 96
5.6.1 NIST 2011 LRE 97
5.6.2 QCRI Arabic DRE 103
5.6.3 RATS LRE 105
5.7 Conclusions 108
5.8 Appendix I 108
5.9 References 109

6 Speaker age estimation using a fusion of the i-vector and NFA framesworks 113
6.1 Abstract 114
6.2 Introduction 114
6.3 Age Estimation from Speech 116
6.3.1 Baseline Approaches 116
6.4 System Description 117
6.4.1 Regression using LSSVR 117
6.4.2 Utterance Modeling 119
6.4.3 System Architecture 121
6.5 Experimental Setup 123
6.5.1 Database 123
6.5.2 Performance Metric 123

6.6 Results and Discussion 124
6.6.1 Baseline Systems Results 124
6.6.2 NFA Framework 125
6.6.3 Proposed Method 125
6.7 Conclusions 127
6.8 Acknowledgements 127
6.9 References 127

7 Normalized ordinal distance **133**
7.1 Abstract 134
7.2 Introduction 134
7.3 Problem Formulation 135
7.3.1 Ordinal Classification 136
7.3.2 Probabilistic-Ordinal classification 136
7.3.3 Partial-Ordinal Classification 136
7.4 Conventional Performance Metrics 137
7.4.1 Mean Zero-One Error (E_{mzo}) 137
7.4.2 Mean Absolute Error of Consecutive Integer Labels ($E^{\text{cil}}_{\text{ma}}$) 137
7.4.3 Percentage of Correctly Fuzzy Classified Instances (P_{cfci}) 138
7.4.4 Average Deviation (E_{ad}) 138
7.4.5 Average Ranked Probability Scores (E_{rps}) 138
7.5 Proposed Performance Metric 139
7.5.1 Ordinal Distance (OD) 139
7.5.2 Normalized Ordinal Distance (E^{p}_{nod}) 140
7.6 Results and Discussion 143
7.6.1 Cumulative Probability Mass Distribution 144
7.6.2 Order of categories 145
7.6.3 Number of Categories 145
7.6.4 Relation to ranked probability score 147
7.6.5 Partial-Ordinal Problems 149
7.7 Conclusion 150
7.8 Acknowledgements 151
7.9 References 151

8 Conclusion **155**
8.1 Contributions 155
8.2 Future Research Directions 156
8.2.1 Signal Representation 156
8.2.2 NFA for phonotactic language recognition 157
8.2.3 Calibration and fusion in ordinal classification problems 157
8.2.4 Adaptation to different applications 158
8.3 References 158

Bibliography **159**

List of Publications **161**

List of Figures

1.1 *A simplified human speech production model and recording channel.* 3
1.2 *Extracting acoustic features from a speech signal and fitting a GMM to them [40].* 5
1.3 *Adapting UBM to an utterance.* 6
1.4 *The block-diagram of the model-based speaker characterization approach in training and testing phases.* 13
1.5 *The block-diagram of the template-based speaker characterization approach in training and testing phases.* 14

2.1 *The block diagram of the proposed speaker age estimation approach in training and testing phases.* 33
2.2 *Age histogram of telephone speech utterances for NIST 2010 and 2008 SRE Databases.* 34
2.3 *The E_{ma} of female and male speakers' age estimation using the proposed method and baseline systems versus target dimension.* 38
2.4 *Pearson correlation coefficient between estimated and true age of female and male speakers using the proposed method and baseline systems versus target dimension.* 39
2.5 *The E_{ma} of female and male speakers' age estimation using the proposed method and Prior baseline system versus the test utterance length.* 41
2.6 *Pearson correlation coefficient between estimated and true age of female and male speakers using the proposed method versus the test utterance length.* 42
2.7 *Age histogram of English and non-English speakers in the NIST* 2008 *SRE database.* 43

3.1 *The block-diagram of the proposed method in the training phase.* 56
3.2 *The block-diagram of the proposed method in testing phases.* . . 57
3.3 *Age histogram of male speakers in the evaluation corpus.* 58

3.4 *Age histogram of female speakers in the evaluation corpus.* . . . 59
3.5 *The MAE of age estimation using the proposed method and NMF versus target dimension.* . . . 60

4.1 *The block diagram of the accent recognition systems in training and testing phases.* . . . 70

5.1 *The adapted weights of the UBM with three Gaussians using the ML method.* . . . 92
5.2 *The space of possible adapted weights of a UBM with three Gaussians using NMF.* . . . 93
5.3 *The space of possible adapted weights of a UBM with three Gaussians using SMM.* . . . 94
5.4 *The space of possible adapted weights of a UBM with three Gaussians using NFA.* . . . 95
5.5 *The histogram of objective function value after convergence for 100 randomly initialized NFA factorizations.* . . . 96
5.6 *The block-diagram of applied classification scheme NIST 2011 LRE and QCRI Arabic DRE experiments.* . . . 98
5.7 *The C_{llr} of language recognition using the proposed method and baseline systems versus subspace vector dimension.* . . . 100
5.8 *The required computation time for estimating the subspace matrices using the proposed method and baseline systems versus subspace vector dimension.* . . . 101
5.9 *The C_{llr} of language recognition using the proposed method and baseline systems in different utterance length conditions.* . . . 102
5.10 *The block-diagram of utterance modeling in intermediate-level fusion.* . . . 102

6.1 *The block-diagram of the proposed speaker age estimation approach in training phase.* . . . 122
6.2 *The block-diagram of the proposed speaker age estimation approach in development and testing phases.* . . . 122
6.3 *Age histogram of telephone speech utterances for NIST 2010 and 2008 SRE Databases.* . . . 126

7.1 *Diagram of unit circle using Minkowski and Ordinal distances of orders 1, 2 and infinity.* . . . 141
7.2 *The effect of using cumulative mass distribution.* . . . 144

List of Tables

2.1 *The E_{ma} (in years) and ρ of male and female speakers' age estimation using SVR and LSSVR.* 36
2.2 *The average E_{ma} (in years) of male and female speakers' age estimation for the baseline systems using* MFCC_{26D} *and* MFCC_{60D} *feature vectors.* 37
2.3 *The average ρ of male and female speakers' age estimation for the baseline systems using* MFCC_{26D} *and* MFCC_{60D} *feature vectors.* 38
2.4 *The E_{ma} and ρ for both English and non-English test sets.* 41
2.5 *The mean and the standard deviation of age estimation absolute error using i-vector-SVR and GMM-PCA-R over male and female utterances.* 44

3.1 *Age group recognition accuracy in %.* 58

4.1 *The number of utterances and speakers for each accent category.* 71
4.2 *Comparison of various i-vector, GWS and GMS based systems. The results are given in P_{cc} and $C_{\text{llr}}^{\text{min}}$.* 73
4.3 *Comparison of NBC, SVM and SRC after feature level fusion. The results are given in P_{cc} and $C_{\text{llr}}^{\text{min}}$.* 73
4.4 *The confusion matrix of accent recognition for i-vector-GWS-GMS-SRC system. The results are given in percentage* 74

5.1 *The C_{llr} of language recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors.* . 103
5.2 *The number of utterances for each dialect category in the QCRI corpus.* 104
5.3 *The number of utterances in different durations in the QCRI corpus.* 104
5.4 *The E_{ic} of dialect recognition using the proposed method and baseline systems in QCRI Arabic DRE experiment (%).* 105

5.5 *The E_{ic} of dialect recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors in QCRI Arabic DRE experiment (%).* 106
5.6 *The number of utterances for each category in the RATS corpus.* 106
5.7 *The E_{ic} of dialect recognition using the proposed method and baseline systems in RATS LRE experiment (%).* 107

6.1 *The E_{ma} (in years) and ρ of male and female speakers' age estimation for the baseline systems.* 125
6.2 *The E_{ma} (in years) and ρ of speakers' age estimation for NFA framework in different subspace dimensions.* 126
6.3 *The E_{ma} (in years) and ρ of speakers' age estimation for estimators 1 and 2.* 127

7.1 The performance of two classifiers measured by E_{mzo}, E_{ad}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞} in example 1. 145
7.2 The performance of two classifiers measured by E_{mzo}, E_{ad}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞} in example 2. 146
7.3 The performance of two classifiers measured by E_{mzo}, E_{ad}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞} in example 3. 148
7.4 Test set datapoints and their corresponding classifier outputs in example 4. 149
7.5 The performance of two classifiers measured by E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞} in example 5. 149
7.6 The output of applied classifiers in example 6. 150
7.7 The performance of two classifiers measured by P_{cfci}, E_{ad}, E_{rps}, E_{nod}^{1}, E_{nod}^{2} and E_{nod}^{∞} in example 6. 151

Chapter 1

Introduction

1.1 Motivations and Goals

Speech signals carry important information about a speaker such as age, gender, language, accent and emotional/psychological state. Automatic identification of speaker characteristics has a wide range of commercial, medical and forensic applications in real-world scenarios [1–6]. For example, in a multilingual call-center, a call should be directed to an agent, whose language matches the customer [2]. To find the best agent for a call, an automatic dialect/accent recognition system can be considered to avoid typical misunderstandings in the agent-customer conversation. In this case, an automatic age estimation can also be applied as elderly customers usually prefer an agent with a slow speech rate [6].

Targeted advertising through the Internet, where user-computer and user-company vocal interaction has increased significantly during the last decades, is another scenario of application. In this case, information about the user's language/accent, age and gender can help to offer appropriate products and services [6].

In video games, knowledge about a user's characteristics can help to adapt the game to him/her. For example, the preference for the game music might differ significantly between a male teenager compared and an adult female.

Speaker characterization is also applied to diagnosis, analysis and monitoring of different diseases such as autism and Parkinson's disease. Different applications of speech technology in medical scenarios are reported in [7–14].

Automatic identification of speaker characteristics can improve the performance of automatic speech recognition (ASR) systems. A fundamental challenge of using ASR systems in real world markets such as telephone networks and personal computers is their significant performance drop for non-native speakers [15, 16]. Consequently, accent/dialect recognition systems can be applied to avoid this problem.

Speaker profiling is also required in many forensic scenarios [3]. Law enforcement agencies have been concerned about different biometric techniques to confirm the identity of an individual. Different biometric characteristics can be used for forensic identification such as fingerprint patterns, face characteristics, hand geometry, signature dynamics and voice patterns. Choosing a method depends on its reliability in a particular application and the available data. In some criminal cases, available evidence might be in the form of recorded conversations. Speech patterns can include important information for law enforcement personnel [3]. For example, a person's speech pattern can provide information about his/her age, gender, dialect, emotional or psychological state and membership of a particular social or regional group. Therefore, speech can be used for speaker identification which is highly demanded in many cases such as kidnapping, threatening calls and false alarms [3].

This research aims to develop accurate methods and tools to identify different characteristics of the speakers. Due to lack of required databases, among all characteristics of speakers, our experiments cover gender recognition, age estimation, language recognition and accent/dialect recognition. However, similar approaches and techniques can be applied to identify other characteristics such as emotional/psychological state.

1.2 Ethical Issues

Similar to other biometric technologies, there are serious ethical issues in the use of speaker characterization technology concerning the personal privacy and the use of personal data. This technology has the capability to limit personal freedom, privacy, anonymity and democratic rights. Therefore, civil liberty organizations and the public have to be seriously concerned about the use of these technologies. Academics, lawyers and civil liberty organizations play an important role to develop workable and deployable approaches to use this technology in a safe and secure manner.

1.3 Challenges

Although experimental studies reveal different acoustic/linguistic cues for each characteristic, the relation of these cues to the target characteristic is usually complex and affected by many other factors such as speech content, language, ethnicity, and emotional condition [17–19]. These issues make automatic speaker profiling very challenging for both humans and machines [17, 20, 21].

Figure 1.1, which shows a simplified model for human speech production, helps to display the underlying difficulties in speaker characterization. In this problem, the recorded speech signal is the only available information and the task is to identify the speaker's characteristics without any information about the articulatory system inputs, other physical and psychological states of the articulatory system and channel characteristics.

Technical factors such as available speech duration, environment, recording device and channel conditions also influence the identification accuracy. In other words, in a typical practical scenario, the quality of the available speech signal and the recording conditions are not controlled and the duration of the speech signal may vary from a few seconds to several hours.

1.4 Related Work

Different approaches have been developed to identify speaker characteristics during the last decades. The first works on this field started in the early 1970s [22, 23]. However, it remains a challenging task due to similarities of acoustic phonetics, phonotactics, and prosodic cues across different characteristics. Furthermore, in many practical cases we have no control over the available speech duration, channel characteristics, and noise level.

Speaker characterization approaches can be divided into phonotactic and acoustic approaches [24]. A phone recognizer followed by language models (PRLM) and parallel PRLM (PPRLM) techniques developed within the language recognition area, are successful phonotactic methods focusing on phone sequences

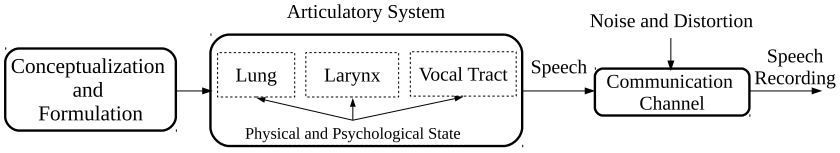

Figure 1.1: *A simplified human speech production model and recording channel.*

as important information of different speaker characteristics such as language, accent/dialect, belonging to a particular social/regional group and even to an age category [25]. Phonotactic approaches are not useful for identification of paralinguistic speaker characteristics such as smoking habit and height. Phonotactic features and acoustic (spectral and/or prosodic) features provide complementary cues, state-of-the-art methods usually apply a combination of both through a fusion of their output scores [24].

The acoustic approaches, which are the main focus of this thesis, enjoy the advantages of requiring no specialized language knowledge [24]. This type of approaches, which can also be applied to identify paralinguistic speaker characteristics, have been widely used in different speaker characterization problems [24, 26–30]. For example, in [31–33] different types of acoustic features and support vector machines (SVM) have been used for speaker age group recognition. In [34], Gaussian mixture model (GMM) mean supervectors and SVM were applied. In the field of speaker recognition, recent advances using i-vectors have increased the recognition accuracy considerably [35]. An i-vector is a compact representation of an utterance in the form of a low-dimensional feature vector. The same idea was also effectively applied to spoken language recognition [36].

Annual paralinguistic challenges held at INTERSPEECH provide a forum for state-of-the-art methods in speaker characterization such as emotional state and age recognition [6, 29]. In these challenges, GMM mean supervectors [37], GMM weight supervectors [38], Maximum-Mutual-Information (MMI) training [30], Joint Factor Analysis (JFA) [30] and fuzzy SVM modeling [39] have been suggested to enhance acoustic modeling quality. A summary of the submitted approaches to these challenges and the obtained results can be found in [6].

1.5 Problem Formulation

In the speaker profiling problem, we are given a training dataset $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \ldots, (\mathcal{X}_s, y_s), \ldots, (\mathcal{X}_S, y_S)\}$, where $\mathcal{X}_s$ denotes the s^{th} utterance of the training dataset, and y_s denotes a label vector that shows the correct label of the utterance (the speaker characteristic that we aim to identify). The goal is to approximate a function (g), such that for an unseen observation $\mathcal{X}^{\text{tst}}$, $\hat{y} = g(\mathcal{X}^{\text{tst}})$ is as close as possible to the true label.

The solutions of this problem can be categorized into model-based and template-based approaches, which are described in Sections 1.6 and 1.7 respectively.

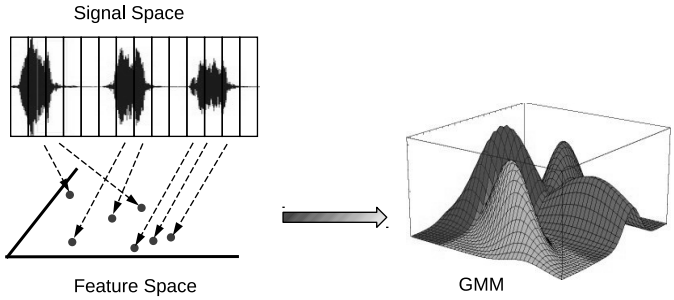

Figure 1.2: *Extracting acoustic features from a speech signal and fitting a GMM to them [40].*

1.6 Model-Based Speaker Characterization

In this category of speaker characterization approaches, to approximate function g, we first convert variable-duration speech signals into fixed-dimensional vectors suitable for classification/regression algorithms, which is usually performed by finding the parameters of a statistical model for the speech signals.

1.6.1 Statistical Modeling and Signal Representation

To convert variable-duration speech signals into fixed-dimensional vectors, a probability density function (PDF) is fitted to acoustic features extracted from the speech signals such that the parameters of the fitted PDF to an utterance characterize the speaker. Since the distribution of acoustic features is complex, a GMM is applied to model the distribution of the acoustic features. Figure 1.2 shows the underlying idea of fitting a GMM to the acoustic features extracted from an utterance.

Due to the lack of data, fitting a separate GMM-based acoustic model to a short utterance cannot be performed accurately, especially in the case of GMMs with a high number of Gaussians. Therefore, parametric utterance adaptation methods are usually applied to adapt a universal background model (UBM) to characteristics of utterances in training and testing databases as shown in Figure 1.3. A UBM has the following data likelihood function $\mathcal{X} =$

$\{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_\tau\}$:

$$p(\mathbf{x}_t|\lambda) = \sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$$

$$\lambda = \{b_c, \mu_c, \mathbf{\Sigma}_c\}, \ c = 1, \ldots C, \tag{1.1}$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$ is a Gaussian probability density function with mean μ_c and covariance matrix $\mathbf{\Sigma}_c$, and C is the total number of Gaussians in the mixture. The parameters of the UBM $-\lambda-$ are estimated on a large amount of training data including many speakers of different characteristics. The parameters of each adapted GMM (Gaussian weights, means and covariances) characterize the corresponding utterance. Different methods have been suggested for Gaussian mean and Gaussian weight adaptation, which are briefly introduced as follows:

Mean Adaptation

In this research, the UBM mean and the adapted mean of the c^{th} Gaussian are denoted by μ_c and $\mathbf{m}_c$ respectively. The main approaches to adapt Gaussian means are

• *Maximum likelihood re-estimation* [41]
In this method, the adapted GMM Gaussian means are estimated by maximizing the likelihood of Eq. 1.1 for the adaptation data over the Gaussian means. Since it is not clear which training sample contributes to which Gaussian, this optimization is challenging. Therefore, rather than directly maximizing the log-likelihood of Eq. 1.1, the auxiliary function of Eq. 1.2, namely complete-data log-likelihood is introduced and an iterative Expectation-Maximization (EM) algorithm is applied [42]. In each E-step of this algorithm, given the predecessor

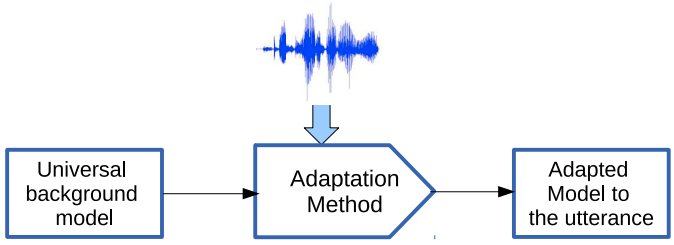

Figure 1.3: *Adapting UBM to an utterance.*

estimate of the model parameters (starting from UBM parameters), the auxiliary function of Eq. 1.2 is formed by calculating the occupation counts $\gamma_{c,t}$ (Eq. 1.3) for all mixture components. In the M-step, model parameters are updated by maximizing the auxiliary function found on the E-step. It is shown that the maximization of the auxiliary function over the model parameters (mean, covariance and weights), increases the data likelihood of Eq. 1.1 [42]. The new model is then considered as the initial model in the next iteration and this iterative process is continued until convergence. The new model in each step is obtained by maximizing the auxiliary function of Eq. 1.2.

$$\Phi(\lambda, \mathbf{m}_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log \left[b_c p(\mathbf{x}_t | \mathbf{m}_c, \mathbf{\Sigma}_c) \right], \tag{1.2}$$

where $\gamma_{c,t}$ is the occupation count for the c^{th} mixture component and the t^{th} segment. Occupation counts are calculated as follows:

$$\gamma_{c,t} = \frac{b_c p(\mathbf{x}_t | \mu_c, \mathbf{\Sigma}_c)}{\sum_{c=1}^{C} b_c p(\mathbf{x}_t | \mu_c, \mathbf{\Sigma}_c)} \tag{1.3}$$

Finally, the adapted means $\mathbf{m}_c$ after the first EM iteration, which are found by maximizing the auxiliary function, are obtained as follows:

$$\mathbf{m}_c = \frac{\sum_{t=1}^{\tau} \mathbf{x}_t \gamma_{c,t}}{\sum_{t=1}^{\tau} \gamma_{c,t}} \tag{1.4}$$

As can be interpreted from Eq. 1.4, a Gaussian mean is updated if its corresponding phonetic context is covered in the adaptation utterance. Consequently, maximum likelihood re-estimation approach does not lead to an accurate adapted model for short utterances.

• *Maximum-a-posteriori* (*MAP*) [43]
Maximum-a-posteriori (MAP) is another approach of Gaussian means adaptation. This method involves a two-step estimation process similar to that of the maximum likelihood re-estimation method. The first steps of MAP and maximum likelihood re-estimation are identical. However, in the second step of the MAP algorithm, the obtained sufficient statistics estimated in the first step are combined with the statistics of the prior mixture parameters using a mixing coefficient η^{μ} controlling the balance between the prior and new information. In this approach, the adapted means $\mathbf{m}_c$ after the first iteration are obtained

as follows:

$$\mathbf{m}_c = \frac{\gamma_c \mathbf{m}_c^* + \eta^\mu \mu_c}{\gamma_c + \eta^\mu} \tag{1.5}$$

$$\mathbf{m}_c^* = \frac{\sum_{t=1}^{\tau} \mathbf{x}_t \gamma_{c,t}}{\sum_{t=1}^{\tau} \gamma_{c,t}} \tag{1.6}$$

$$\gamma_c = \sum_{t=1}^{\tau} \gamma_{c,t} \tag{1.7}$$

As can be interpreted from Eq. 1.5, the mixtures with high posterior probabilities rely more on the new data and mixtures with low posterior probabilities rely more on the prior distribution.

• *Maximum likelihood linear regression* (*MLLR*) [44]
MLLR is a speaker adaptation approach assuming the following linear transformation that maps the UBM means to their speaker-adapted equivalents.

$$\mathbf{m}_c = \mathcal{W} \begin{bmatrix} \mu_c \\ 1 \end{bmatrix} = \mathcal{W} \mu_c^+, \tag{1.8}$$

where $\mathcal{W}$ is a transformation matrix of proper size estimated based on the maximum likelihood criterion such that the following auxiliary function is maximized.

$$\Phi(\lambda, \mathcal{W}) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log b_c p(\mathbf{x}_t | \mathcal{W} \mu_c^+, \mathbf{\Sigma}_c), \tag{1.9}$$

In the case of short utterances, the elements of $\mathcal{W}$ might be poorly estimated. Consequently, we will have a poor mapping, which leads to an inaccurate adapted model [45, 46]. Therefore, it is too risky to use conventional MLLR for short utterances.

• *The i-vector framework* [35]
The i-vectors framework [35] developed within the field of speaker recognition is a another Gaussian mean adaptation approach. This method assumes that the adapted Gaussian means supervector, which is obtained by extraction and concatenation of Gaussian means, can be decomposed as

$$\mathbf{m} = \mathbf{u} + \mathbf{T}\mathbf{v}, \tag{1.10}$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean

offset $\mathbf{Tv}$. In this framework, $\mathbf{T}$ and $\mathbf{v}$ are estimated using the EM algorithm. In the E-step, $\mathbf{T}$ is assumed to be known, and we update $\mathbf{v}$. Similarly in the M-step, $\mathbf{v}$ is assumed to be known and we try to update $\mathbf{T}$.

The vector $\mathbf{v}$ is treated as a latent variable with the standard normal prior and the i-vector is its MAP point estimate which is obtained by maximization of the following auxiliary function over $\mathbf{v}$

$$\Omega(\lambda, \mathbf{v}) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log b_c p(\mathbf{x}_t | \left[\mu_c + \mathbf{T}_c \mathbf{v}\right], \mathbf{\Sigma}_c) \mathcal{N}(\mathbf{v}), \tag{1.11}$$

where $\mathcal{N}(\mathbf{v})$ denotes the standard normal distribution of $\mathbf{v}$ and $\mathbf{T}_c$ are the rows of the subspace matrix $\mathbf{T}$, which correspond to the c^{th} Gaussian mean. In the E step, the posterior distribution of $\mathbf{v}$ is Gaussian with the following mean $\mathbf{v}_\mu$ and covariance matrices $\mathbf{v}_\sigma$ [47]:

$$\mathbf{v}_\sigma = \left[\mathbf{I} + \sum_c \gamma_c \mathbf{T}_c^{'} \bar{\mathbf{\Sigma}}_c^{-1} \mathbf{T}_c\right]^{-1} \tag{1.12}$$

$$\mathbf{v}_\mu = \mathbf{v}_\sigma \sum_c \left[\mathbf{T}_c^{'} \bar{\mathbf{\Sigma}}_c^{-1} \sum_t \gamma_{c,t} (\mathbf{x}_t - \mathbf{m}_c)\right], \tag{1.13}$$

where $\mathbf{I}$ denotes an identity matrix of appropriate size and $\mathbf{m}_c$ and $\bar{\mathbf{\Sigma}}_c$ are adapted mean and covariance of the c^{th} Gaussian, which are updated during each EM iteration starting from UBM parameters.

In the M-step, the subspace matrix $\mathbf{T}$ is estimated via maximization of the following auxiliary function over $\mathbf{T}$

$$\widetilde{\Omega}(\lambda, \mathbf{T}) = \sum_{s=1}^{S} \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t,s} \log b_{c,s} p(\mathbf{x}_{t,s} | \left[\mu_c + \mathbf{T}_c \mathbf{v}_s\right], \mathbf{\Sigma}_{c,s}). \tag{1.14}$$

The procedure for training $\mathbf{T}$ can be found in [47].

In this approach, the i-vectors are the low-dimensional representation of an audio recording that can be used for classification and estimation purposes.

Weight Adaptation

In this research, the UBM weight and the adapted weight of the c^{th} Gaussian are denoted by b_c and w_c respectively. The main approaches to adapt Gaussian weights are

• *Maximum likelihood re-estimation* [41]
The maximum likelihood re-estimation approach is also applied to adapt Gaussian weights in a similar way as Gaussian means. In this method, the auxiliary function of Eq. 1.2 is maximized over w_c. Since $p(x_t|\mu_c, \mathbf{\Sigma}_c)$ remains unchanged in this maximization process, the auxiliary function Eq. 1.2 can be simplified to

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c, \tag{1.15}$$

Finally, the adapted weights w_c after the first EM iteration are obtained as follows:

$$w_c = \frac{1}{\tau} \sum_{t=1}^{\tau} \gamma_{c,t} \tag{1.16}$$

• *Maximum-a-posteriori* (*MAP*) [43]
We can apply the MAP approach to adapt Gaussian weights in a similar way as Gaussian means. In this method, the adapted weights w_c after the first iteration are obtained as follows:

$$w_c = \left[\frac{\gamma_c w_c^* + \mathfrak{y}^b b_c}{\gamma_c + \mathfrak{y}^b} \right] \wp \tag{1.17}$$

$$w_c^* = \frac{1}{\tau} \sum_{t=1}^{\tau} \gamma_{c,t}, \tag{1.18}$$

where $\mathfrak{y}^b$ is a mixing coefficient controlling the balance between the prior and new information and $\wp$ is a scaling factor to ansure the obtained adapted weights sum up to unity. As can be interpreted from Eq. 1.17, the mixtures with high posterior probabilities rely more on the new data and mixtures with low posterior probabilities rely more on the prior distribution.

• *Non-negative matrix factorization* (*NMF*) [48]
The main assumption of the NMF based method [48] is that for a given utterance,

$$w_c = \mathbf{B}_c \mathbf{h}, \tag{1.19}$$

where $\mathbf{B}_c$ is a non-negative row vector forming the c^{th} row of the non-negative subspace matrix $\mathbf{B}$, and $\mathbf{h}$ is a low-dimensional and non-negative vector representing the utterance. In this method, $\mathbf{B}_c$ and $\mathbf{h}$ are initialized randomly, and then updated in an alternating method [49] to maximize the objective function Eq. 1.15. The adapted GMM weights are constrained to be non-negative and sum up to one. Since all elements of subspace matrix $\mathbf{B}$, and

subspace vector $\mathbf{h}$ are non-negative, the adapted weights using NMF are also non-negative. To keep the sum of adapted GMM weights equal to one, the columns of subspace matrix $\mathbf{B}$ are normalized to sum up to one after updating it in each iteration. This normalization is also performed for the subspace vector $\mathbf{h}$. Details of this parameter re-estimation method can be found in [48].

The subspace matrix $\mathbf{B}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{h}$ for each utterance in train and test datasets.

• *Subspace multinomial model* (*SMM*) [50]
Kockmann *et al.* introduced the SMM approach for Gaussian weight adaptation and decomposition with application to prosodic speaker verification [50]. The main assumption of this method is that for a given utterance,

$$w_c = \frac{\exp(z_c + \mathbf{A}_c\mathbf{q})}{\sum_{j=1}^{C}\exp(z_j + \mathbf{A}_j\mathbf{q})}, \tag{1.20}$$

where z_c is the c^{th} element of the origin of the supervector subspace, $\mathbf{A}_c$ is the c^{th} row of the subspace matrix and $\mathbf{q}$ is a low-dimensional vector representing the utterance.

In this method, $\mathbf{A}_c$ and $\mathbf{q}$ are estimated using a two-stage iterative algorithm similar to EM to maximize the objective function (1.15). For each stage of the EM-like algorithm, an iterative optimization approach similar to that of the Newton-Raphson scheme is applied. Details of this parameter re-estimation approach, which involves calculation of the Hessian matrix and estimating the subspace vectors, can be found in [50].

The subspace matrix $\mathbf{A}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{q}$ for each utterance in train and test datasets.

Interpretation of the adapted Gaussian means and weights

Assuming the UBM represents the acoustic space of the sufficiently large training dataset, adapted Gaussian means show the pronunciation type of different speech components, e.g. phonemes. However, many speech components are unobserved —or weakly observed— in adaptation utterance, hence their corresponding Gaussian means remain unchanged during the adaptation process. Consequently, they do not contribute in speaker characterization process appropriately.

Adapted Gaussian weights indicate the existence level of the corresponding speech components in the adaptation utterance, i.e. the weights of unobserved Gaussians are zero and they increase as the observations (existence levels) of corresponding speech components in the adaptation utterance rise. Therefore,

despite of Gaussian means, the weights of unobserved or weakly observed Gaussians contribute in speaker characterization process by carrying information about the existence level of the corresponding speech components.

During the last decade, research in the field of speaker/language recognition focused on Gaussian means [24, 51] However, Gaussian weights carry complementary information to the Gaussian means and applying it may improve the identification accuracy. This is among the main intentions of this work.

Session Variability Compensation

Session compensation is one of the most dominant topics in the speaker recognition field [35, 52]. The main reason of using session compensation techniques is removing different session variabilities from the feature vectors (such as GMM supervectors or i-vectors) to allow the subsequent modeling approaches to better observe important between-class information. In the context of speaker characterization, session variation is anything that makes features corresponding to speakers of the same target characteristic appear different such as phonetic content, transmission channel, recording device and emotional state. Two widely used approaches for session variability compensation are linear discriminant analysis (LDA) [53] and within-class covariance normalization (WCCN) [54]. LDA provides a transformation such that the ratio of the transformed between-class-scatter and the transformed within-class-scatter is maximized. WCCN transforms the within-class covariance of the vector space to an identity matrix [54]. In doing so, directions of relatively high within-class variation will be attenuated, and thus prevented from dominating the space [54].

Training and Testing

The principle of the model-based speaker characterization approach is illustrated in Figure 1.4. As it can be interpreted from this figure, in the training phase, each utterance in the training dataset is converted to a vector representing the corresponding utterance. Then, a session variability compensation approach is applied to remove the session variability as described in Section 1.6.1. Finally, the obtained vectors along with their corresponding label (target characteristic of speaker) are used to train the classifier/regression algorithm. In the testing phase, the same utterance representation approach applied in the training phase is used to model the utterance of an unseen speaker. Then, the trained session compensation method is used to remove the session variability. Finally, the trained classifier/regression algorithm uses the obtained vector to identify speaker characteristic.

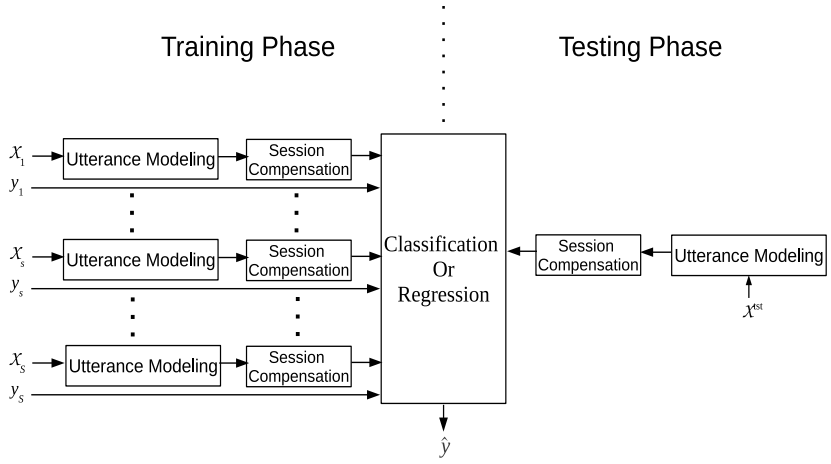

Figure 1.4: *The block-diagram of the model-based speaker characterization approach in training and testing phases.*

1.7 Template-Based Speaker Characterization

In these category of speaker characterization approaches, function g is approximated directly from acoustic features extracted from the speech signals, i.e. there is no statistical model between acoustic/prosodic features and classifier/regression algorithm. The block-diagram of this type of methods is shown in figure 1.5. An example of using this approach can be found in [17], where classification and regression trees (CART) is applied to recognize speaker age group from acoustic and prosodic features. In [55] a Bayesian classifier is applied for speaker age group recognition from acoustic features.

The most important advantage of this category of speaker characterization methods is its conceptual simplicity compared to model-based approaches. However, depending on the acoustic/prosodic feature extraction frame-size and length of training utterances, the number of input-output patterns can be very high, which increases the computation time dramatically.

1.8 Summary of Contributions

In this thesis, we focus on model-based approaches and design new tools and techniques to identify different characteristics of speakers from speech signals.

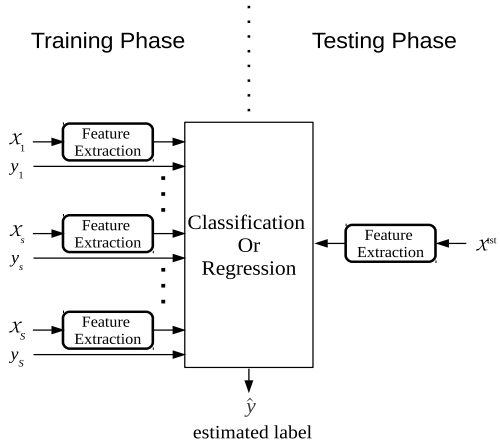

Figure 1.5: *The block-diagram of the template-based speaker characterization approach in training and testing phases.*

The main contributions of this thesis are summarized as follows:
1. Designing a new i-vector-based approach for speaker age estimation, which improves the accuracy of the state-of-the-art speaker age estimation methods with statistical significance[1].
2. Analyzing the effect of major factors influencing the automatic age estimation systems.
3. Exploring the availability of information in GMM weights by applying them to speaker gender detection, age estimation and native language recognition problems. Our experiments show that GMM weights carry less, yet complementary, information to Gaussian means and i-vectors.
4. Proposing a new subspace approach for GMM weight adaptation, namely non-negative factor analysis (NFA). Motivated by the results of our experiments indicating that GMM weights carry complementary information to Gaussian means, we developed the NFA framework to use this information effectively. NFA applies a constrained factor analysis and suggests a new low-dimensional utterance representation approach based on Gaussian weights.
5. Proposing an intermediate-level fusion of the i-vector and the NFA frameworks to improve the recognition accuracy of state-of-the-art i-vector-based approach in language and dialect recognition tasks.
6. Proposing a hybrid architecture of i-vector and NFA frameworks for speaker age estimation, which improves state-of-the-art i-vector based system

[1]This method was the winner of International Speech Communication Association best student paper award at INTERSPEECH 2012

considerably.

7. Introducing Ordinal Distance of two arbitrary vectors in Euclidean space and proposing an application independent performance metric, namely normalized ordinal distance, for ordinal, probabilistic-ordinal and partial-ordinal classification problems based on the defined ordinal distance. OD and NOD can be applied in identification of many speaker characterization problems with ordinal nature such as age group recognition, identifying the level of intoxication and height group estimation.

1.9 Outline of the Thesis

The dissertation is organized in 8 chapters based on author's published, accepted and submitted peer-reviewed papers during the course of this project.

Chapter 2 introduces a new approach for speaker age estimation based on the i-vector framework. In this method, each utterance is modeled by its corresponding i-vector. Then, a Within Class Covariance Normalization (WCCN) [54] technique is applied for session variability compensation. Finally, least squares support vector regression (LSSVR) is applied to estimate the age of speakers. The proposed method is trained and tested on telephone conversations of the National Institute of Standards and Technology (NIST) 2010 and 2008 speaker recognition evaluations databases. In this Chapter, we apply tests of significance to ensure the effectiveness of the proposed scheme compared to conventional methods. We also investigate the impact of major speech or speaker related factors influencing the automatic age estimation systems in a typical practical case, namely the available speech sample duration and the spoken language.

Chapter 3 proposes a new gender detection and age recognition technique. The proposed method in **Chapter 2** and all baseline systems were developed using Gaussian means. While a GMM is characterized by Gaussian means, covariances and weights, state-of-the-art systems usually do not use weights or covariances. In **Chapter 3** our aim is to test the available information in Gaussian weights and its effectiveness in the case of speaker age and gender recognition. In this method, speakers are modeled by their corresponding hidden Markov model (HMM) weight supervectors. Then, weighted supervised non-negative matrix factorization (WSNMF) is applied to recognize the gender-age group of speakers and to reduce the dimension of the input HMM weight supervectors. Finally, a LSSVR is employed to estimate the age of speakers using the obtained low-dimensional vectors. Evaluation results on a corpus of read and spontaneous speech in Dutch, namely N-best, confirms the effectiveness of the proposed scheme.

In **Chapter 4** three utterance modeling approaches, namely Gaussian mean supervector, i-vector and Gaussian weight supervector, are applied to the native language recognition (L1-recognition) problem. For each utterance modeling method, three different classifiers, namely the support vector machine (SVM), the naive Bayesian classifiers (NBC) and the sparse representation classifiers(SRC), are employed to find out suitable matches between the utterance modeling schemes and the classifiers. Our experiments in this chapter are performed using English utterances of speakers, whose native language is Russian, Hindi, American English, Thai, Vietnamese and Cantonese. These utterances are drawn from the National Institute of Standards and Technology (NIST) 2008 Speaker Recognition Evaluation (SRE) database. It is shown that the concatenation of i-vectors, GMM mean supervectors, and GMM weight supervectors improve the accuracy of accent recognition compared to each of them separately.

An extensive comparison of weight supervectors, mean supervectors and i-vectors is performed in **Chapter 4**. Our experiments in this chapter show that Gaussian weights, which entail a lower dimension compared to Gaussian mean supervectors, carry less, yet complementary, information to GMM means. Inspired from the results of chapters **4** and **3** we tried to improve the effectiveness of Gaussian weight based systems by introducing a new subspace method for GMM weight adaptation based on a factor analysis similar to that of the i-vector framework. **Chapter 5** introduces a non-negative factor analysis (NFA) approach for GMM weight decomposition and adaptation. This modeling suggests a new low-dimensional utterance representation method, which uses a factor analysis similar to that of the i-vector framework. The obtained subspace vectors are then applied in conjunction with i-vectors to the language/dialect recognition problem. The suggested approach is evaluated on the NIST 2011 and RATS language recognition evaluation (LRE) corpora and on the QCRI Arabic dialect recognition evaluation (DRE) corpus. In this chapter, the proposed GMM weight subspace vectors are fused with i-vectors effectively to form new vectors representing the utterances. The experimental results show that the proposed fusion improves the performance of the state-of-the-art i-vector framework for the language and dialect recognition tasks.

Motivated by the success of the NFA framework, in **Chapter 6** we have introduced a hybrid architecture of the NFA approach and the i-vector frameworks for speaker age estimation problem. Evaluation on NIST 2010 and 2008 SRE corpora, show that the proposed hybrid architecture improves the results of the i-vector framework considerably.

Chapter 7 introduces a new performance metric for ordinal, probabilistic-ordinal and partial-ordinal classification problems. An important drawback in many speaker characterization problems such as age group recognition,

identifying the level of intoxication and height group estimation is that there is an intrinsic ordering between the classification categories. Measurement of classification performance in this type of problems, namely ordinal classification, is challenging and conventional performance metrics such as error rate, cost of log-likelihood ratio and mean squared error do not reflect the effectiveness of the classifier appropriately. To solve this problem, **Chapter 7** introduces a new performance metric for ordinal classification problems, namely normalized ordinal distance (E^p_{nod}). This performance metric is conceptually simple, computationally inexpensive and application-independent. The advantages of the proposed method over the conventional approaches and its different characteristics are shown using several numerical examples.

The thesis ends with a conclusion in **Chapter 8**.

1.10 References

[1] Y. Muthusamy, E. Barnard, and R. Cole, "Reviewing automatic language identification," *Signal Processing Magazine, IEEE*, vol. 11, no. 4, pp. 33–41, 1994.

[2] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[3] D. C. Tanner and M. E. Tanner, *Forensic aspects of speech patterns: voice prints, speaker profiling, lie and intoxication detection*. Lawyers and Judges Publishing, 2004.

[4] G. Dobry, R. M. Hecht, M. Avigal, and Y. Zigel, "Supervector Dimension Reduction for Efficient Speaker Age Estimation Based on the Acoustic Speech Signal," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 7, pp. 1975–1985, 2011.

[5] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism," in *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*, 2013.

[6] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. MuLler, and S. Narayanan, "Paralinguistics in speech and language—state-of-the-art and the challenge," *Computer Speech and Language*, vol. 27, no. 1, pp. 4–39, 2013.

[7] J. Schoentgen, "Vocal cues of disordered voices: an overview," *Acta acustica united with acustica*, vol. 92, no. 5, pp. 667–680, 2006.

[8] Y. Harrison and J. A. Horne, "The impact of sleep deprivation on decision making: a review.," *Journal of Experimental Psychology: Applied*, vol. 6, no. 3, p. 236, 2000.

[9] S. Sapir, L. O. Ramig, J. L. Spielman, and C. Fox, "Formant centralization ratio: a proposal for a new acoustic measure of dysarthric speech," *Journal of Speech, Language and Hearing Research*, vol. 53, no. 1, p. 114, 2010.

[10] D. Oller, P. Niyogi, S. Gray, J. Richards, J. Gilkerson, D. Xu, U. Yapanel, and S. Warren, "Automated vocal analysis of naturalistic recordings from children with autism, language delay, and typical development," *Proceedings of the National Academy of Sciences*, vol. 107, no. 30, pp. 13354–13359, 2010.

[11] A. A. Dibazar, S. Narayanan, and T. W. Berger, "Feature analysis for automatic detection of pathological speech," in *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, vol. 1, pp. 182–183, IEEE, 2002.

[12] N. Malyska, T. F. Quatieri, and D. Sturim, "Automatic dysphonia recognition using biologically inspired amplitude-modulation features," in *Proc. ICASSP*, vol. 1, pp. 873–876, 2005.

[13] A. Maier, T. Haderlein, U. Eysholdt, F. Rosanowski, A. Batliner, M. Schuster, and E. Noth, "Peaks–a system for the automatic evaluation of voice and speech disorders," *Speech Communication*, vol. 51, no. 5, pp. 425–437, 2009.

[14] I. Rektorova, J. Barrett, M. Mikl, I. Rektor, and T. Paus, "Functional abnormalities in the primary orofacial sensorimotor cortex during speech in parkinson's disease," *Movement disorders*, vol. 22, no. 14, pp. 2043–2051, 2007.

[15] A. Hanani, "Human and computer recognition of regional accents and ethnic groups from british english speech," *University of Birmingham*, July 2012.

[16] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," *Columbia University*, 2011.

[17] S. Schotz, *Perception, analysis and synthesis of speaker age*, vol. 47. Citeseer, 2006.

[18] M. H. Bahari and H. Van hamme, "Speaker age estimation and gender detection based on supervised Non-Negative Matrix Factorization," in *Proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pp. 1–6, 2011.

[19] M. H. Bahari and H. Van hamme, "Speaker age estimation using Hidden Markov Model weight supervectors," in *11th IEEE Int. Conf. Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, 2012.

[20] T. Bocklet, A. Maier, and E. Noth, "Age determination of children in preschool and primary school age with GMM-based supervectors and support vector machines regression," in *Proc. Text, Speech and Dialogue*, pp. 253–260, 2008.

[21] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[22] R. G. Leonard and G. R. Doddington, "Automatic Language Identification.," *Technical Report RADC-TR-74-2007TI-347650, RADC/Texas Instruments, Inc., Dalas, TX*, 1974.

[23] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *The Journal of the Acoustical Society of America*, vol. 62, p. 708, 1977.

[24] A. Hanani, M. Russell, and M. Carey, "Human and computer recognition of regional accents and ethnic groups from british english speech," *Computer Speech and Language*, vol. 27, no. 1, pp. 59–74, 2013.

[25] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[26] F. Biadsy, J. Hirschberg, and D. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," in *Proc. Interspeech*, 2011.

[27] A. Demarco and S. Cox, "Iterative classification of regional british accents in i-vector space," in *Proc. Machine Learning in Speech and Language Processing*, 2012.

[28] G. Dobry, R. Hecht, M. Avigal, and Y. Zigel, "Dimension reduction approaches for SVM based speaker age estimation," in *Proc. Interspeech*, pp. 2031–2034, 2009.

[29] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Proc. Interspeech*, pp. 2794–2797, 2010.

[30] M. Kockmann, L. Burget, and J. Cernocky, "Brno University of Technology System for Interspeech 2010," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2822–2825, 2010.

[31] D. Mahmoodi, A. Soleimani, H. Marvi, F. Razzazi, M. Taghizadeh, and M. Mahmoodi, "Age estimation based on speech features and support vector machine," in *3rd Computer Science and Electronic Engineering Conference*, pp. 60–64, 2011.

[32] C.-C. Chen, P.-T. Lu, M.-L. Hsia, J.-Y. Ke, and O.-C. Chen, "Gender-to-Age hierarchical recognition for speech," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, pp. 1–4, 2011.

[33] C. van Heerden, E. Barnard, M. Davel, C. van der Walt, E. van Dyk, M. Feld, and C. Muller, "Combining regression and classification methods for improving automatic speaker age recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 5174–5177, 2010.

[34] T. Bocklet, A. Maier, J. Bauer, F. Burkhardt, and E. Noth, "Age and gender recognition for telephone applications based on GMM supervectors and support vector machines," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1605–1608, 2008.

[35] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[36] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, pp. 857–860, 2011.

[37] T. Bocklet, G. Stemmer, V. Zeissler, and E. Noth, "Age and Gender Recognition Based on Multiple Systems Early vs. Late fusion," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2830–2833, 2010.

[38] R. Porat, D. Lange, and Y. Zigel, "Age recognition based on speech signals using weights supervector," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2814–2817, 2010.

[39] P. Nguyen, T. Le, D. Tran, X. Huang, and D. Sharma, "Fuzzy Support Vector Machines for Age and Gender Classification," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2806–2809, 2010.

[40] N. Dehak and S. Shum, "Low-dimensional speech representation based on factor analysis and its applications," *Interspeech*, 2011.

[41] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[42] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[43] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of markov chains," *Speech and audio processing, ieee transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[44] C. J. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[45] M. H. Bahari and H. Van hamme, "Speaker adaptation using maximum likelihood general regression," in *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pp. 29–34, IEEE, 2012.

[46] M. Bahari and H. Van hamme, "Rapid speaker adaptation using maximum likelihood neural regression," in *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2011.

[47] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, no. 5, pp. 980–988, 2008.

[48] X. Zhang, K. Demuynck, and H. Van hamme, "Rapid Speaker Adaptation in Latent Speaker Space with Non-negative Matrix Factorization," *Speech Communication*, 2013.

[49] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[50] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Cernockỳ, "Prosodic speaker verification using subspace multinomial models with

intersession compensation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[51] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[52] M. McLaren and D. van Leeuwen, "Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 20, no. 3, pp. 755–766, 2012.

[53] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification and Scene Analysis 2nd ed.," 1995.

[54] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, vol. 4, 2006.

[55] N. Minematsu, M. Sekiguchi, and K. Hirose, "Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 137–140, 2002.

Chapter 2

Age estimation from telephone speech using i-vectors

This chapter is based on the following articles:
1) Bahari, M.H., McLaren, M., Van hamme, H., van Leeuwen D., (2014), "Speaker age estimation using i-vectors," Engineering Applications of Artificial Intelligence, Elsevier (Accepted).
2) Bahari, M.H., McLaren, M., Van hamme, H., van Leeuwen D. (2012), "Age estimation from telephone speech using i-vectors," 13th Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 506-509, USA (**International Speech Communication Association Best Student Paper Award**)

2.1 Abstract

In this chapter, a new approach for age estimation from speech signals based on i-vectors is proposed. In this method, each utterance is modeled by its corresponding i-vector. Then, a Within-Class Covariance Normalization technique is used for session variability compensation. Finally, a least squares support vector regression (LSSVR) is applied to estimate the age of speakers. The proposed method is trained and tested on telephone conversations of the National Institute for Standard and Technology (NIST) 2010 and 2008 speaker recognition evaluation databases. Evaluation results show that the proposed method yields significantly lower mean absolute estimation error and higher Pearson correlation coefficient between chronological speaker age and estimated speaker age compared to different conventional schemes. Finally, the effect of some major factors influencing the proposed age estimation system, namely utterance length and spoken language are analyzed.

2.2 Introduction

Speech signals carry important information about the speaker such as gender, age, language, dialect, emotional or psychological state. In this research, we focus on speaker age estimation, which has a wide range of commercial applications such as interactive voice response systems, targeted advertising, service customization, and natural human-machine interaction [1]. Speaker age estimation also plays an important role in directing the investigation process in many forensic cases such as kidnapping, threatening calls, and false alarms [2].

Experimental studies reveal major effects of vocal aging on the speech signal such as lowered speaking rate and increased jitter and shimmer [3], and has shown to negatively influence speaker recognition performance [4]. However, the relation of these acoustic cues with speaker age is usually complex and affected by many other factors such as speech content, language, gender, weight, height, emotional condition, smoking and drinking habits [3, 5, 6]. Furthermore, in many practical cases we have no control over the available speech duration, content, language, etc.. These issues make automatic speaker age estimation very challenging for both humans and machines [3, 7, 8].

Figure 1.1, which shows a simplified model for human speech production, helps to display the underlying difficulties in speaker age estimation. In this problem, the recorded speech signal is the only available information and the task is to estimate one of the physical states of the articulatory system, namely the speaker's age, without any information about the system inputs, channel

characteristics and the other psychological and physical states of the articulatory system such as gender, emotional state and smoking habit.

Technical factors such as available speech duration, environment, recording device and channel conditions also influence the estimation accuracy. In other words, in a typical practical scenario, the quality of the available speech signal and the recording conditions are not controlled and the duration of the speech signal may vary from a few seconds to several hours.

2.2.1 Related Work

Studies on the influence of ageing on voice started in the late 1950s [9]. However, the first automatic speaker age recognition systems were developed around four decades later in the early 2000s [10–13]. During this decade, many different techniques, mostly inspired from the automatic speaker and language recognition fields, have been suggested for categorizing speakers based on their age groups. For example, using different types of acoustic features and Support Vector Machines (SVM) [14–16], Gaussian Mixture Model (GMM) mean supervectors and SVM [17], nuisance attribute projection [18], anchor models [18] and parallel phoneme recognizers [19]. The age sub-challenge of the Interspeech 2010 paralinguistic challenge provided a forum for presenting state-of-the-art methods in speaker age group classification [20]. Participants of the age sub-challenge tried to categorize speakers of telephony data in the "aGender" corpus into four age groups — 7 to 14 (Child), 15 to 24 (Youth), 25 to 54 (Adult) and 55 to 80 (Senior) years old. In this sub-challenge, GMM mean supervectors [21], GMM weight supervectors [22], Maximum-Mutual-Information (MMI) training [23] and fuzzy SVM modeling [24] have been suggested to enhance acoustic modeling quality. A brief overview of different proposed methods in this sub-challenge is presented in [8], which also introduces an age group recognition approach using acoustic and prosodic level information fusion.

In speaker age group recognition, crisp borders are assumed between different age groups. For example, in the mentioned age sub-challenge, a speaker with age 54 belongs to the adult group and a 55 year old speaker belongs to the senior category, These two speakers who have only one year of age difference and share many similarities are considered to be from two different categories, while a 80 year old speaker with distant characteristics is in the same category as the 55 year old speaker. This setup causes many problems in training, testing, and performance measurement. To avoid these troubles, recently it has been suggested to use regression for age estimation [1, 5–7, 25]. A probabilistic interpretation of the posterior distribution of age estimation and its calibration is presented in [26].

2.2.2 Motivations, Goals and Summary of Contributions

One effective approach to age estimation from speech involves modeling speech recordings with Gaussian Mixture Model (GMM) mean supervectors to use them as features in Support Vector Regression (SVR) [1, 7]. Similar Support Vector Machine (SVM) techniques have been successfully applied to different speech processing tasks such as speaker recognition [27]. While effective, GMM mean supervectors are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. Consequently, dimension reduction through PCA-based methods has been found to improve performance in age estimation from GMM mean supervectors [1]. In the field of speaker recognition, recent advances using so-called i-vectors [28] have increased the classification accuracy considerably. An i-vector is a compact representation of an utterance in the form of a low-dimensional feature vector. The same idea was also applied in speaker or language and dialect recognition effectively [29, 30]. In [31], we successfully replaced GMM mean supervectors by low-dimensional i-vectors to model utterances in an SVR based speaker age estimation system. The results of evaluation on the NIST 2010 and 2008 SRE databases illustrated that the i-vector based speaker age estimator increases the estimation accuracy.

In this chapter, we extended our previous work by
1. Applying Within Class Covariance Normalization (WCCN) [32] technique for session variability compensation. In [31], we have applied WCCN to normalize utterances of each age group. This method was not successful. In this chapter we updated our strategy to use WCCN for normalizing utterances of each speaker rather than age group.
2. Replacement of SVR by least squares SVR (LSSVR) to improve the computational cost.
3. Updating the evaluation setup such to increase the size of training dataset, which helps the classifier to observe more variability of the data.
4. Using standard z-test to analyze the statistical significance of the obtained improvement by the proposed method.
5. Investigate the effect of utterance length on the proposed automatic speaker age estimation system.
6. Investigate the language mismatch on the proposed method.

The rest of this chapter is organized as follows. In Section 2.3 the problem of speaker age estimation and different conventional approaches addressing this issue are described. In section 2.4, the proposed approach is elaborated. Section 2.5 explains our experimental setup. The evaluation results and an investigation of parameters affecting speaker age estimation are presented and discussed in section 2.6. The chapter ends with conclusions in section 2.7.

2.3 Age Estimation from Speech

In speaker age estimation, we are given a training dataset of speech recordings $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \ldots, (\mathcal{X}_s, y_s), \ldots, (\mathcal{X}_S, y_S)\}$. In this set, $\mathcal{X}_s$ and y_s denote the s^{th} utterance of the training dataset and its corresponding speaker age, respectively. The goal is to design an estimator function g, such that for an utterance of an unseen speaker $\mathcal{X}^{\text{tst}}$, the actual speaker age is predicted accurately.

2.3.1 Baseline Approaches

In this chapter, we use three baseline approaches with which we compare our proposed regression techniques:
Prior: The most basic choice for the estimator function is the average age of the training data, $g(x^{\text{tst}}) = \frac{1}{S}\sum_s y_s$. This estimator, labeled as *prior* in the rest of this chapter, intuitively provides a reference level of accuracy.
GMM-R: Different methods have been introduced to reach an effective speaker age estimation [1]–[5]. For example, Bocklet *et al.* introduced GMM-R to estimate the age of children from GMM mean supervectors derived from their utterances [7]. Given an utterance, Maximum A Posteriori adaptation (MAP) is applied to adapt a Universal Background Model (UBM) to the speech characteristics of the speaker [27]. The component means of the obtained GMM are then extracted and concatenated to form a GMM mean supervector representing the utterance. Finally, an SVR is applied as a function approximator to estimate the speakers' age.
GMM-PCA-R and **GMM-WPPCA-R:** The approach of GMM-R was adopted and extended by Dobry *et al.* [1] by applying dimension reduction techniques to the supervector. Methods such as Principal Component Analysis (PCA) and Weighted-Pairwise PCA (WPPCA) were applied and investigated. It was concluded that WPPCA, which is a supervised dimensionality reduction approach working based on nuisance attribute projection [1], yields more accurate results. These speaker age estimators, labeled GMM-PCA-R and GMM-WPPCA-R, are used as contrastive baseline systems in this chapter.

2.4 Age Estimation using i-vectors

This section briefly describes the main components of the i-vector based age estimation approach, namely SVR and LSSVR, the i-vector framework and WCCN. Then, the proposed method is elaborated and finally the proposed scheme is presented.

2.4.1 Regression

In this section, SVR and LSSVR are briefly introduced.

Support Vector Regression

Support Vector Regression (SVR) is a function approximation approach developed as a regression version of the widely known classification paradigm, namely Support Vector Machines (SVM) [33, 34]. While SVMs perform the classification task by determining the maximum margin separation hyperplane between two classes, SVRs carry out the regression task by finding the optimal regression hyperplane in which most of training samples lie within an ϵ-margin around this hyperplane [34]. In a typical regression problem a training dataset $S^{\text{tr}} = \{(\mathbf{w}_1, y_1), \ldots, (\mathbf{w}_s, y_s), \ldots, (\mathbf{w}_S, y_S)\}$ is given, where $\mathbf{w}_s$ denotes a vector of observed features of the data item and y_s denote model input and corresponding output of the s^{th} data point respectively. The objective of the regression analysis is to determine a function $f(\mathbf{w})$, so as to predict the desired outputs accurately. In the primal form of SVR the following relation is considered for $f(\mathbf{w})$:

$$f(\mathbf{w}) = \varpi' \Phi(\mathbf{w}) + z \tag{2.1}$$

where $\Phi(\mathbf{w})$ denotes a mapping function in the feature space, ϖ is a row vector with the same dimension of $\Phi(\mathbf{w})$, $z \in \mathbb{R}$ is a constant and and $'$ represents the transpose operator. Using Vapnik's ϵ-insensitive loss function the model training—estimation of ϖ and z—is formulated as to minimize

$$\frac{1}{2} \|\varpi\|^2 + \lambda \sum_{s=1}^{S} (\xi_s + \xi_s^*) \tag{2.2}$$

subject to

$$\begin{cases} y_s - \varpi' \Phi(\mathbf{w}_s) - z & \leq \epsilon + \xi_s \\ \varpi' \Phi(\mathbf{w}_s) + z - y_s & \leq \epsilon + \xi_s^* \\ \xi_s, \xi_s^* & \geq 0. \end{cases} , \tag{2.3}$$

where ξ_s and ξ_s^* are slack variables vanishing during the optimization process, $\epsilon > 0$ controls the ϵ-insensitive zone used for fitting the training data and $\lambda > 0$ determines the trade-off between the flatness of $f(a)$ and the cost of tolerating deviations larger than ϵ.

For high dimensional data, this constrained minimization problem can be solved more efficiently by introducing a dual set of variables and solving the following dual optimization problem [34]

$$\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{m,s=1}^{S}(\alpha_s-\alpha_s^*)(\alpha_m-\alpha_m^*)\langle\Phi(\mathbf{w}_s),\Phi(\mathbf{w})\rangle$$

$$-\epsilon\sum_{s=1}^{S}(\alpha_s+\alpha_s^*)+\sum_{s=1}^{S}(\alpha_s-\alpha_s^*)y_s, \tag{2.4}$$

subject to the constraints

$$\begin{cases} \sum_{s=1}^{S}(\alpha_s-\alpha_s^*)=0 \\ 0\le\alpha_s\le\lambda, \qquad n=1,\cdots,N\ , \\ 0\le\alpha_s^*\le\lambda, \qquad n=1,\cdots,N \end{cases} \tag{2.5}$$

where $\langle\cdot,\cdot\rangle$ describes the dot product and α and α^* are the dual set of variables. The resulting SVR model is

$$f(\mathbf{w})=\sum_{s=1}^{S}\beta_s\langle\Phi(\mathbf{w}_s),\Phi(\mathbf{w})\rangle+z \tag{2.6}$$

$$=\sum_{s=1}^{S}\beta_s K(\mathbf{w}_s,w)+z, \tag{2.7}$$

where $K(\mathbf{w}_s,w)$ is the kernel function. Any function meeting the Mercer's condition can be used as the kernel function [33, 34]. Parameters $\beta_s=\alpha_s-\alpha_s^*$ are calculated through solving the dual optimization problem and have the following relation to ϖ

$$\varpi=\sum_{s=1}^{S}\beta_s\Phi(\mathbf{w}_s). \tag{2.8}$$

Since both the primal and dual optimization problem are convex, a unique optimal solution can be found efficiently using numerical methods such as quadratic programming (QP) [34]. Computing parameters β_s and z is explained in [34] in detail.

In the baseline systems GMM-PCA-R and GMM-WPPCA-R [1], SVR model training and testing is implemented using LIBSVM [35] and the hyper-parameters of the SVR such as the minimal error margin ϵ and error cost

factor λ are tuned using the N-fold cross validation technique on the training dataset. In this research, we use the same toolbox and apply the same approach to tune the hyper-parameters.

Least Squares Support Vector Regression

Least Squares Support Vector Machine (LSSVM), which is a variant of SVM, was introduced by Suykens and Vandewalle [36]. It is employed as a machine learning tool for classification, clustering and regression tasks. Compared to SVM, LSSVM benefits from a faster training process because the quadratic programming problem of SVM is reduced to that of solving a system of linear equations. Furthermore, the LSSVM formulation involves fewer tuning parameters [37]. A continuous function can be fitted to the training data with a Least Squares Support Vector Regressor (LSSVR), a technique which shares many of the advantages of LSSVM classification. In primal form of LSSVR, which is the same as SVR, the following relation is considered for $f(\mathbf{w})$

$$f(\mathbf{w}) = \varpi' \Phi(\mathbf{w}) + z. \tag{2.9}$$

In LSSVR, a least squares loss function is applied instead of Vapnik's ϵ-insensitive loss function to simplify the formulations to minimize

$$\frac{1}{2} \|\varpi\|^2 + \frac{1}{2} \vartheta \sum_{s=1}^{S} e_s^2 \tag{2.10}$$

subject to

$$y_s = \varpi' \Phi(\mathbf{w}_s) + z + e_s, \tag{2.11}$$

where ϑ is a error cost factor playing the same role of λ in the SVR formulation and $e_s \in \mathbb{R}$ are error variables.

Similar to SVR, for high dimensional data this optimization problem can be solved more efficiently by introducing the Lagrangian variables ν and solving the following dual optimization problem [36]

$$\Psi(\varpi, z, e, \nu) = \frac{1}{2} \|\varpi\|^2 + \frac{1}{2} \vartheta \sum_{s=1}^{S} e_s^2 \tag{2.12}$$

$$- \sum_{s=1}^{S} \nu_s \{\varpi' \Phi(\mathbf{w}_s) + z + e_s - y_s\}. \tag{2.13}$$

One can solve this optimization problem directly by taking the partial derivative of Ψ with respect to ϖ, z, e and ν and setting the results to zero which leads

to solving a linear system of equations. Inserting the obtained results in 2.9 leads to the regression function

$$f(\mathbf{w}) = \sum_{s=1}^{S} \nu_s \langle \Phi(\mathbf{w}_s), \Phi(\mathbf{w}) \rangle + z \tag{2.14}$$

$$= \sum_{s=1}^{S} \nu_s K(\mathbf{w}_s, \mathbf{w}) + z, \tag{2.15}$$

where $K(\mathbf{w}_s, \mathbf{w})$ is the kernel function and ν and z are the solution to optimization problem 2.12.

LSSVR has two advantages and one drawback compared to SVR. The first advantage of LSSVR is that its model training is faster as its dual form corresponds to solving a linear system which involves less computation time compared to a QP problem of SVR. The second advantage is that the LSSVR is faster to tune as its formulation involves fewer hyperparameters to tune (the minimal error margin ϵ is not used here). A drawback of this simplification is the loss of sparseness (ν is less sparse compared to β), which has been highlighted in literature [38, 39].

In this research, the LSSVR models training and testing is implemented using LSSVMlab [36] and the Hyperparameters of the LSSVR are tuned on the training set using the N-fold cross validation technique.

2.4.2 The i-vector framework

The age estimation approaches described in section 2.3.1 are based on GMM mean supervectors and have been shown to yield reasonable performance. In the related field of speaker recognition, GMM supervectors are commonplace. Recent progress in this field, however, has found an alternate method of modeling GMM supervectors that provides far superior speaker recognition performance [28]. This technique, referred to as i-vector framework, assumes the GMM mean supervector, $\mathbf{m}$, that best represents a set of features in an utterance can be decomposed as

$$\mathbf{m} = \mathbf{u} + \mathbf{T}\mathbf{v} \tag{2.16}$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace (400 dimensions in this work) and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean offset $\mathbf{T}\mathbf{v}$. The vector $\mathbf{v}$ is commonly referred to as the i-vector and has a standard normal distribution. Subspace $\mathbf{T}$ is estimated via factor analysis to represent the directions that best separate different speech recordings in a large development dataset. An efficient procedure for training

T and MAP adaptation of i-vectors **v** can be found in [40]. In this approach, i-vectors are the low-dimensional representation of an audio recording that can be used for classification and regression purposes.

2.4.3 i-vector Session Compensation

Session compensation is one of the most dominant topics in the speaker recognition field [28, 41]. The main reason of using session compensation techniques is removing different session variabilities from the feature vectors (such as GMM supervectors or i-vectors) to allow the subsequent modeling approaches to better observe important between-class information. In this chapter, we use Within-Class Covariance Normalization (WCCN) to normalize the within-class covariance of the i-vector space to the identity matrix [32]. In doing so, directions of relatively high within-class variation will be attenuated and thus prevented from dominating the space [32]. The WCCN transformation matrix $\mathbf{B}_{\mathrm{W}}$ is found through Cholesky decomposition of

$$\left[\frac{1}{\jmath}\sum_{j=1}^{\jmath}\frac{1}{N_j}\sum_{i=1}^{N_j}\left(\mathbf{v}_j^i-\bar{\mathbf{v}}_j\right)\left(\mathbf{v}_j^i-\bar{\mathbf{v}}_j\right)'\right]^{-1}=\mathbf{B}_{\mathrm{W}}\mathbf{B}_{\mathrm{W}}', \tag{2.17}$$

where $\mathbf{v}_j^i$ is the i^{th} i-vector in the j^{th} speaker, $\bar{\mathbf{v}}_j=\frac{1}{N_j}\sum_i^{N_j}\mathbf{v}_j^i$ is the mean of the observations for the j^{th} speaker, N_j denotes the number of utterances of the j^{th} speaker and $\jmath$ is the total number of speakers in the training dataset.

2.4.4 Train and Test

The principle of the proposed age estimation approach is illustrated in Figure 2.1. As it can be interpreted from this figure, in the training phase, each utterance in the training dataset is converted to an i-vector. Then, WCCN is used to remove the session variability as described in Section 2.4.3. Finally, the obtained vectors along with their corresponding chronological speaker age are used to train the regressor. In the testing phase, an i-vector is extracted from the utterance of an unseen speaker. Then, WCCN is used to remove the session variability. Finally, the trained regressor uses the obtained vector to estimate the chronological age of test speaker.

The use of i-vectors for age estimation has several distinct advantages over GMM supervectors. Firstly, the relatively low dimensionality of i-vectors (400) significantly reduces the computational burden of model training and estimation compared to a GMM supervector dimensionality of greater than 12,000 used

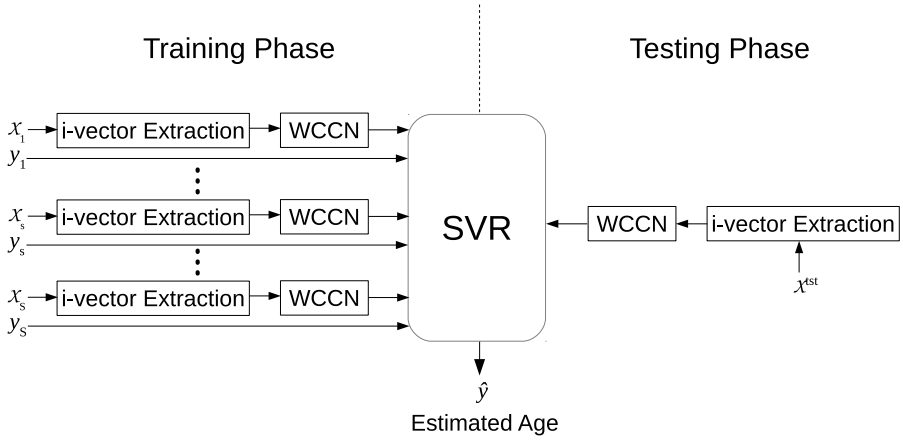

Figure 2.1: *The block diagram of the proposed speaker age estimation approach in training and testing phases.*

in this work. Secondly, subspace adaptation of i-vector $\mathbf{v}$ results in a more reliable estimation of the true model means $\mathbf{m}$ in the context of limited training data [29].

2.5 Experimental Setup

2.5.1 Database

The National Institute of Standards and Technology (NIST) has held annual or biannual Speaker Recognition Evaluations (SRE) for the past two decades. With each SRE, a large database of telephone conversations (and more recently microphone speech) are released along with an evaluation protocol. These conversations typically last five minutes and originate from a large number of participants for whom meta data is recorded—including participant age and language. The NIST databases where chosen for this work due to the large number of speakers meeting the i-vector framework requirement for a considerable amount of development data to estimate subspace matrix $\mathbf{T}$ accurately. In our experiments, first a development dataset is formed, which includes over $30,000$ speech recordings sourced from NIST 2004–2006 SRE databases, to estimate the parameters of UBM and the subspace matrix ($\mathbf{T}$).

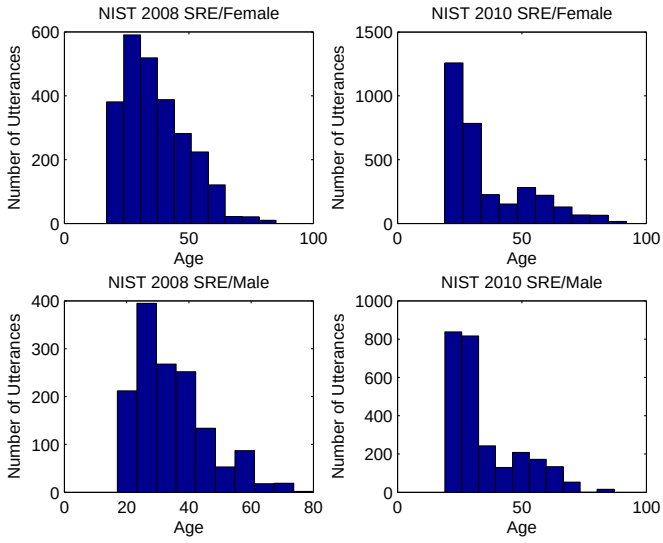

Figure 2.2: *Age histogram of telephone speech utterances for NIST 2010 and 2008 SRE Databases.*

The procedure of obtaining the applied UBM and subspace matrix is presented in [41].

To form the train and test datasets for speaker age estimation, telephone recordings from the common protocols of the NIST 2010 and 2008 SRE corpora are used. The core protocol, short2-short3, from the 2008 database contains 3772 telephone recordings from 1154 speakers for whom the age is between 20 and 70. The language label of 3726 utterances is given in this database. Among these, 2656 utterances are English and the remaining 1070 utterances are from 26 different non-English languages including Russian, Italian and Japanese. Similarly, the extended core-core protocol of the 2010 database contains 5479 telephone speech segments from 422 speakers for whom the age is between 20 and 70. All utterances of this database are English. There is no overlap between speech recordings extracted from the NIST 2010 and NIST 2008 SRE databases.

Figure 2.2 illustrates the age histograms of male and female speakers in the NIST 2010 and 2008 SRE databases.

2.5.2 Performance Metric

The effectiveness of the applied methods is evaluated using the Mean Absolute Error (E_{ma}) of the estimated speakers' age and Pearson's correlation coefficient (ρ) between the chronological speakers' age and the estimated speakers' age. The measure E_{ma} is calculated using:

$$E_{\text{ma}} = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \left| \hat{y}_k - y_k \right|, \tag{2.18}$$

where $\hat{y}_k$ and y_k are the estimated and the chronological age of the k^{th} utterance of the testing dataset respectively. κ is the total number of utterances in the testing dataset. Further,

$$\rho = \frac{1}{\kappa - 1} \sum_{k=1}^{\kappa} \left(\frac{\hat{y}_k - \mu_{\hat{y}}}{\sigma_{\hat{y}}} \right) \left(\frac{y_k - \mu_y}{\sigma_y} \right), \tag{2.19}$$

where $\mu_{\hat{y}}$ and $\sigma_{\hat{y}}$ are the mean and the standard deviation of the speakers' estimated age respectively. Similarly μ_y and σ_y denote the mean and the standard deviation of the speakers' chronological age respectively.

We also apply the standard z-test to analyze the statistical significance level of differences between the mean absolute errors of applied systems.

2.6 Results and Discussion

This section presents the evaluation results of the baseline systems and compares them to the introduced i-vector based age estimation system.

The applied GMM in all experiments consist of 512 mixture components. To study the effect of the acoustic features, two types of feature vectors have been tested for the baseline systems. The first type, labeled MFCC_{26D}, consists of 13 Mel-Frequency Cepstrum Coefficients (MFCCs) including appended energy with their first order derivatives, forming a 26 dimensional acoustic feature vector. The second type, MFCC_{60D}, consists of 20 MFCCs including appended energy with their first and second order derivatives, forming a 60 dimensional acoustic feature vector. In both cases, a hamming window is used and the sampling rate, frame rate, frame size and number of Mel frequency channels are 8000 Hz, 100 Hz, 0.02 s and 30 respectively. To have more reliable features, Wiener filtering, speech activity detection [42] and feature warping [43] have been applied as front-end processing. The former type, MFCC_{26D}, matches the configuration of features applied in [1] and the latter type, MFCC_{60D}, is very common in state-of-the-art i-vector based speaker recognition systems.

Table 2.1: *The E_{ma} (in years) and ρ of male and female speakers' age estimation using SVR and LSSVR.*

Regression Method	Female		Male	
	E_{ma}	ρ	E_{ma}	ρ
SVR 1	7.59	0.80	7.97	0.69
SVR 2	7.48	0.80	7.92	0.70
LSSVR	7.44	0.80	7.87	0.70

2.6.1 SVR and LSSVR

In this section, an experiment is performed to investigate the performances of SVR and LSSVR for regression in this problem and choose the regression method with more accurate estimation results for the rest of the experiments in this chapter.

In this experiment, the NIST 2008 and 2010 SRE databases are used for training and testing respectively and the acoustic features are $\text{MFCC}_{26\text{D}}$. Each utterance in the training and testing datasets is modeled using its corresponding GMM mean supervector. Then, an SVR or an LSSVR are applied as a function approximator to estimate the speakers' age.

Like the baseline systems GMM-PCA-R and GMM-WPPCA-R, SVR model training and testing is performed using LIBSVM [35] and the SVR Hyperparameters ϵ and λ are tuned using the 5-fold cross-validation. Since it is shown in [1] that the radial basis function (RBF) kernel leads to more accurate estimation compared to the linear kernel, we apply the RBF kernel in our experiments. Two methods are applied to determine the width of the Gaussian functions. In the first scheme, which is adopted from [1], the width of the Gaussian functions was set to $\sqrt{\det(\mathbf{\Sigma}_{\text{trn}})/2}$, where $\mathbf{\Sigma}_{\text{trn}}$ is the training feature vectors covariance matrix and $\det(.)$ denotes determinant operator. It was mentioned in [1] that $\sqrt{\det(\mathbf{\Sigma}_{\text{trn}})/2}$ was found to be optimal on a number of empirical experiments. The results of this method, labeled as SVR 1, are listed in the first row of Table 2.1. In the second approach, labeled as SVR 2, the 5-fold cross-validation is used to tune the width of the Gaussian functions.

The applied LSSVR in this experiment also uses the RBF kernel and 5-fold cross-validation is applied to tune its error cost factor and Gaussian width.

Table 2.1 shows the obtained results using SVR 1, SVR 2 and LSSVR in this experiment. This table shows that LSSVR estimates the speakers' age more accurately compared to SVR 1 and SVR 2 in this experiment. LSSVR is selected for the rest of experiments in this chapter rather than conventional SVR due to

Table 2.2: *The average E_{ma} (in years) of male and female speakers' age estimation for the baseline systems using* MFCC_{26D} *and* MFCC_{60D} *feature vectors.*

System Configuration	Female		Male	
	MFCC_{26D}	MFCC_{60D}	MFCC_{26D}	MFCC_{60D}
Prior	10.57	10.57	10.08	10.08
GMM-R	6.19	6.60	6.93	7.53
GMM-PCA-R	6.26	6.21	6.79	6.71
GMM-WPPCA-R	6.25	6.17	6.74	6.74

the obtained marginal improvement and faster and easier model training and tuning.

2.6.2 Baseline Systems Results

In this section, the performances of baseline systems, namely prior, GMM-R, GMM-PCA-R and GMM-WPPCA-R, are investigated.

To evaluate the baseline systems on all available utterances, 15-fold cross-validation is used. Therefore, first all speakers in the NIST 2008 and 2010 SRE databases are divided into 15 disjoint folds. Then, 15 independent experiments are run so that in each experiment, a new fold is used as the testing dataset and the remaining 14 folds are used as training dataset. The average E_{ma} and ρ of male and female speakers' age estimation using the baseline systems in all 15 experiments with both types of acoustic features are listed in tables 2.2 and 2.3 respectively. In this experiment, PCA and WPPCA have been tested over different target dimensions between 100 and 1000. Tables 2.2 and 2.3 only include the best results, which were obtained for target dimensions 300 and 400 for GMM-PCA-R and GMM-WPPCA-R respectively.

Results in tables 2.2 and 2.3 indicate that the GMM-R system is remarkably more accurate than the prior system. This shows that the GMM supervectors contain speaker information including age. The Tables 2.2 and 2.3 also show that the PCA and WPPCA based systems outperform the GMM-R system, thus demonstrating the benefit of dimension reduction of the GMM supervectors prior to regression. Unlike [1] our experiments do not show remarkable advantage for using WPPCA over PCA. It is also interpreted from tables 2.2 and 2.3 that increasing the acoustic dimension from 26 to 60 slightly improves the estimation accuracy for GMM-PCA-R and GMM-WPPCA-R. Therefore, in the rest of our experiments we focused on the second type of acoustic features, MFCC_{60D}.

Table 2.3: *The average ρ of male and female speakers' age estimation for the baseline systems using* $\text{MFCC}_{26\text{D}}$ *and* $\text{MFCC}_{60\text{D}}$ *feature vectors.*

System Configuration	Female		Male	
	$\text{MFCC}_{26\text{D}}$	$\text{MFCC}_{60\text{D}}$	$\text{MFCC}_{26\text{D}}$	$\text{MFCC}_{60\text{D}}$
Prior	0	0	0	0
GMM-R	0.78	0.73	0.69	0.59
GMM-PCA-R	0.77	0.78	0.71	0.72
GMM-WPPCA-R	0.77	0.78	0.71	0.71

2.6.3 i-vectors for Age Estimation

The results of the proposed method for speakers' age estimation are presented in this section.

Figures 2.3 and 2.4 present the E_{ma} of the estimated age and the ρ between the chronological speakers' age and the estimated speakers' age using the proposed method and the baseline systems for different target dimensions respectively. These figures show that the proposed method, labeled i-vector-WCCN-R, is more accurate than the other state-of-the-art approaches. Note that this improvement

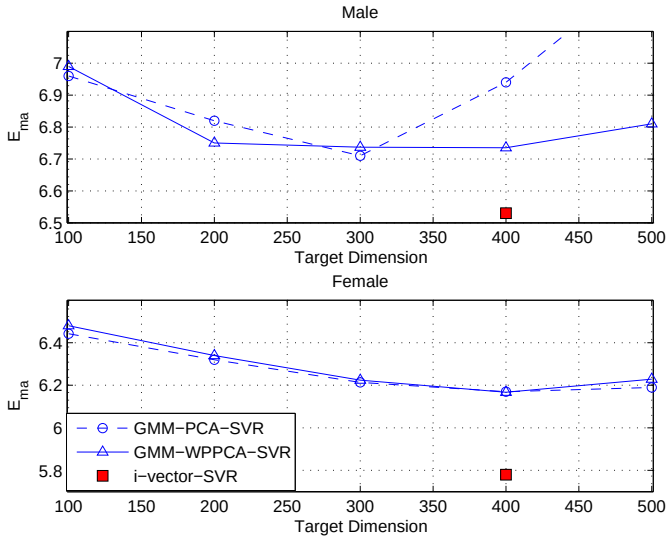

Figure 2.3: *The E_{ma} of female and male speakers' age estimation using the proposed method and baseline systems versus target dimension.*

was obtained without any optimization over the target dimension in the i-vector framework. Therefore, in figures 2.3 and 2.4, the result of proposed method is only shown for dimension 400. In the standard i-vector framework, the optimization over the target dimension is usually very time-consuming and computationally expensive.

The ρ and E_{ma} of age estimation using the proposed approach are 0.772 and 6.08 respectively. Therefore, the proposed method improves ρ by 12.9%, 2.0% and 2.6% relative to GMM-R, GMM-PCA-R and GMM-PCA-R respectively. The E_{ma} is also improved by 41%, 13%, 5% and 4.8% relative to Prior, GMM-R, GMM-PCA-R and GMM-PCA-R respectively. A standard z-test for comparing two means show that the E_{ma} of the i-vector based system method is significantly lower than that of the best baseline system, namely GMM-PCA-R, at the 99% confidence level. Details of this test are presented in Appendix I.

We also investigated using i-vectors without session variability compensation, like our earlier work [31]. In this case, the ρ and E_{ma} are 0.76 and 6.22 respectively. This experiment shows that session variability compensation using WCCN relatively improves the ρ and E_{ma} by 1.5% and 2.2% respectively.

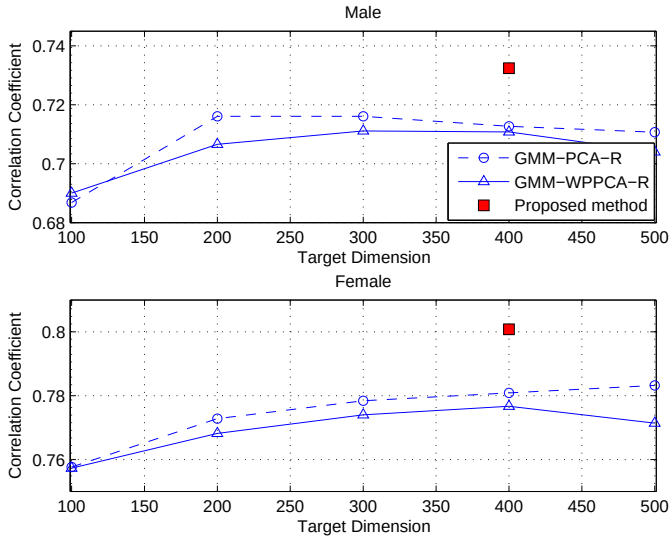

Figure 2.4: *Pearson correlation coefficient between estimated and true age of female and male speakers using the proposed method and baseline systems versus target dimension.*

2.6.4 The Effect of Utterance Length

In a typical practical case, the duration of the available speech sample may vary from a few seconds to several hours. Although there is literature on the effect of available utterance duration on speaker recognition systems [44], there is no published research on this topic for automatic speaker age estimation systems. In this section, we analyze the performance of the proposed i-vector based speaker age estimation system with respect to speech duration in the terms of E_{ma} and ρ.

In this experiment, first all speakers in the NIST 2008 and 2010 SRE databases are divided into 15 disjoint folds. Then, 15 independent experiments are run so that in each experiment, a new fold is used as testing dataset and the rest 14 folds are used as training dataset. Each utterance in the testing dataset typically contains around 80 seconds of active speech. In order to study the effect of test sample duration, we synthesized test datasets of 5, 10, 20 and 40 seconds by truncating the feature streams after speech activity detection. For consistency in our results, the test samples that contained less than 40 seconds of nominal speech using our speech detection algorithm were discarded from all results reported in this experiment. The procedure and details of obtaining corresponding i-vectors for truncated test samples is explained in [45].

The corresponding E_{ma} and ρ values are presented in figures 2.5 and 2.6. The performance of the proposed method decreases as the test utterance duration is reduced. This is more evident when the utterance duration is less than 10 seconds. However, the results of the proposed method remain significantly more accurate than the prior even for the utterances of 5 seconds length.

2.6.5 The Effect of Language

Braun and Cerrato performed a number of experiments to evaluate the ability of human listeners in estimating speakers' age across different languages [46]. They concluded that the age can be estimated almost as accurately when the listeners are familiar with the language of the speaker as when they are not. However, Schotz considered the language as an important source influencing the acoustic analysis of speaker age [3]. Feld *et al.* studied the effect of language mismatch between train database and test samples on automatic speaker age estimation systems. In this section, we analyze the effect of language mismatch on the proposed i-vector based age estimation system.

In this experiment, the train database is NIST 2010 SRE, which includes 5634 English utterances from 445 speakers. There are two test databases in this experiment, the English and non-English parts of the NIST 2008 SRE database.

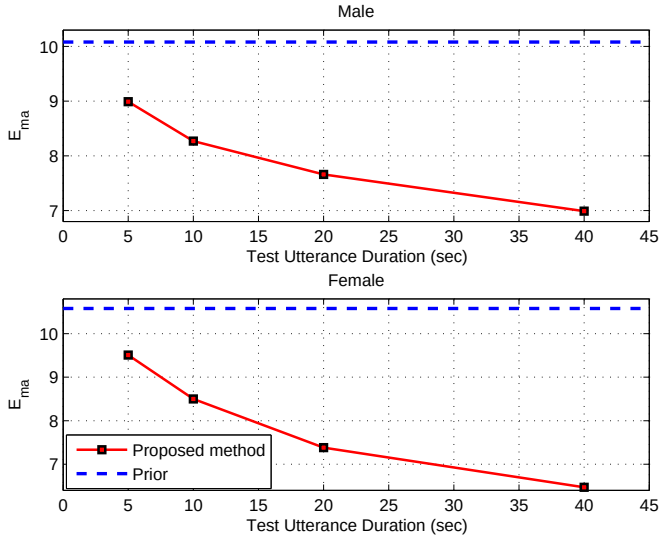

Figure 2.5: *The E_{ma} of female and male speakers' age estimation using the proposed method and Prior baseline system versus the test utterance length.*

Figure 2.7 illustrates the age histograms of the English and non-English speakers of the NIST 2008 SRE database. To eliminate the effect of utterance length, we synthesized test samples of 40 seconds by truncating the feature streams after speech activity detection. The E_{ma} and ρ of this experiment for both English and non-English test sets are listed in table 2.4.

Results in table 2.4 indicate that language mismatch between train database and test samples causes a large performance degradation in both E_{ma} and ρ. It is obvious that the E_{ma} for the English test set is significantly less than that of the non-English test set for both male and female utterances.

Table 2.4: *The E_{ma} and ρ for both English and non-English test sets.*

System Configuration	Female		Male	
	English	Non-English	English	Non-English
E_{ma}	6.92	8	7.72	8.32
ρ	0.66	0.42	0.50	0.32

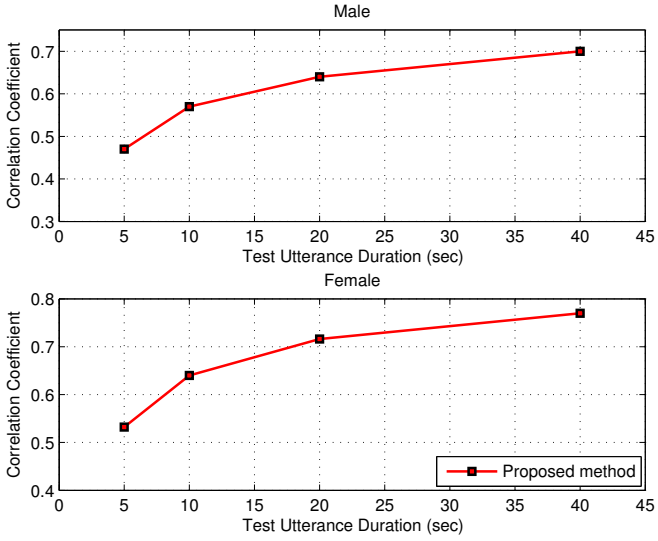

Figure 2.6: *Pearson correlation coefficient between estimated and true age of female and male speakers using the proposed method versus the test utterance length.*

2.7 Conclusions

In this chapter, utterance modeling with i-vectors, which was successfully applied to speaker recognition, has been used in conjunction with a WCCN and a LSSVR to address speaker age estimation. For the evaluation, telephone utterances of NIST 2010 and 2008 SRE databases have been used. Assessment results show that the accuracy of the proposed approach is significantly better than different conventional methods. The experiments on analyzing the effect of utterance duration reveals that the performance of the proposed method degrades as the utterance length decreases especially for samples shorter than 20 seconds. However, it is still more accurate than the prior baseline system even for utterances of 5 seconds in length. Analyzing the effect of language shows that the language mismatch between train and test databases significantly decreases the performance of the age estimation system.

2.8 Appendix I

In this appendix, a statistical analysis is presented to compare the mean absolute errors of age estimation obtained by the i-vector-SVR and GMM-PCA-R.

Since the values of populations variances are unknown, tests for the comparison of two means should be conducted with the a t-test normally. However, both sample sizes are greater than 30 in this case and we can work with the standard normal distribution (z-test) instead of Student distribution (t-test). In the standard z-test for comparison of two means, the z value is calculated as follows:

$$z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.20}$$

where $\bar{x}_1$, s_1, and n_1 denote the mean, the variance and total number of samples in the first set respectively. Similarly, $\bar{x}_2$, s_2, and n_2 are the mean, the variance and sample size in the second set respectively.

In the comparison of the mean absolute errors of age estimation obtained by the i-vector-SVR ($\bar{x}_1$) and GMM-PCA-R ($\bar{x}_2$), the null hypothesis is $\bar{x}_2 \leq \bar{x}_1$ and the alternative hypothesis is $\bar{x}_2 > \bar{x}_1$. With significance levels $\alpha = 0.01$ and

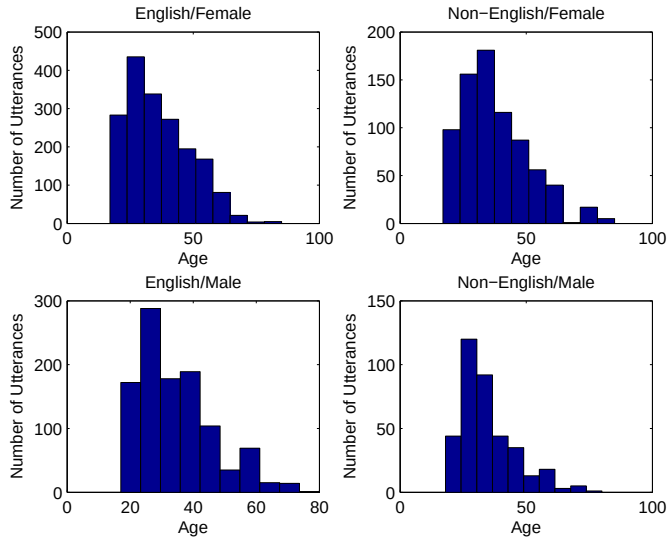

Figure 2.7: *Age histogram of English and non-English speakers in the NIST 2008 SRE database.*

Table 2.5: *The mean and the standard deviation of age estimation absolute error using i-vector-SVR and GMM-PCA-R over male and female utterances.*

Gender	Parameter	Proposed method	GMM-WPPCA-R	z
Male	$\bar{x}_i$	6.53	6.74	1.7
	s_i	5.36	5.54	
	n_i	3883	3883	
Female	$\bar{x}_i$	5.78	6.17	4.13
	s_i	4.78	4.92	
	n_i	5292	5292	
Both	$\bar{x}_i$	6.10	6.41	4.15
	s_i	5.05	5.20	
	n_i	9175	9175	

$\alpha = 0.05$, the critical value of z are 2.33 and 1.645 respectively for a one tail test.

The mean and the standard deviation of age estimation absolute error using i-vector-SVR and GMM-PCA-R over male and female utterances are listed in Table 2.5.

As it is shown in Table 2.5, the obtained z for male and female utterances is greater than the critical value of z for significance levels $\alpha = 0.05$ and $\alpha = 0.01$ respectively. Therefore, the null hypnosis is rejected and it is concluded that the alternative hypnosis is true.

In the test of significance, we are trying to compare GMM-WPPCA-R and the proposed method. Consequently, all results of the proposed method (regardless of gender) can be considered in one class and all the results of GMM-WPPCA-R are assumed to be in the other class. The last row of Table 2.5 shows the mean and the standard deviation of age estimation absolute error using the proposed method and GMM-WPPCA-R over all utterances regardless of gender (labeled both). The obtained z value of this experiment is 4.15 which is greater than the critical value of z for significance level $\alpha = 0.01$.

2.9 References

[1] G. Dobry, R. M. Hecht, M. Avigal, and Y. Zigel, "Supervector Dimension Reduction for Efficient Speaker Age Estimation Based on the Acoustic Speech Signal," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 7, pp. 1975–1985, 2011.

[2] D. C. Tanner and M. E. Tanner, *Forensic aspects of speech patterns: voice prints, speaker profiling, lie and intoxication detection*. Lawyers and Judges Publishing, 2004.

[3] S. Schotz, *Perception, analysis and synthesis of speaker age*, vol. 47. Citeseer, 2006.

[4] F. Kelly, A. Drygajlo, and N. Harte, "Speaker verification in score-ageing-quality classification space," *Computer Speech and Language*, 2013.

[5] M. H. Bahari and H. Van hamme, "Speaker age estimation and gender detection based on supervised Non-Negative Matrix Factorization," in *Proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pp. 1–6, 2011.

[6] M. H. Bahari and H. Van hamme, "Speaker age estimation using Hidden Markov Model weight supervectors," in *11th IEEE Int. Conf. Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, 2012.

[7] T. Bocklet, A. Maier, and E. Noth, "Age determination of children in preschool and primary school age with GMM-based supervectors and support vector machines-regression," in *Proc. Text, Speech and Dialogue*, pp. 253–260, 2008.

[8] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[9] E. D. Mysak, "Pitch and duration characteristics of older males," *Journal of Speech, Language and Hearing Research*, vol. 2, no. 1, p. 46, 1959.

[10] S. E. Linville, *Vocal aging*. Singular Thomson Learning, 2001.

[11] C. Muller, F. Wittig, and J. Baus, "Exploiting speech for recognizing elderly users to respond to their special needs," in *Proc. 8th European Conf. Speech Communication and Technology (Eurospeech)*, pp. 1305–1308, 2003.

[12] N. Minematsu, M. Sekiguchi, and K. Hirose, "Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. I–137, 2002.

[13] I. Shafran, M. Riley, and M. Mohri, "Voice signatures," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 31–36, 2003.

[14] D. Mahmoodi, A. Soleimani, H. Marvi, F. Razzazi, M. Taghizadeh, and M. Mahmoodi, "Age estimation based on speech features and support vector machine," in *3rd Computer Science and Electronic Engineering Conference*, pp. 60–64, 2011.

[15] C.-C. Chen, P.-T. Lu, M.-L. Hsia, J.-Y. Ke, and O.-C. Chen, "Gender-to-Age hierarchical recognition for speech," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, pp. 1–4, 2011.

[16] C. van Heerden, E. Barnard, M. Davel, C. van der Walt, E. van Dyk, M. Feld, and C. Muller, "Combining regression and classification methods for improving automatic speaker age recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 5174–5177, 2010.

[17] T. Bocklet, A. Maier, J. Bauer, F. Burkhardt, and E. Noth, "Age and gender recognition for telephone applications based on GMM supervectors and support vector machines," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1605–1608, 2008.

[18] G. Dobry, R. Hecht, M. Avigal, and Y. Zigel, "Dimension reduction approaches for SVM based speaker age estimation," in *Proc. Interspeech*, pp. 2031–2034, 2009.

[19] F. Metze, J. Ajmera, R. Englert, U. Bub, F. Burkhardt, J. Stegmann, C. Muller, R. Huber, B. Andrassy, J. Bauer, and B. Littel, "Comparison of four approaches to age and gender recognition for telephone applications," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. IV–1089, 2007.

[20] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Proc. Interspeech*, pp. 2794–2797, 2010.

[21] T. Bocklet, G. Stemmer, V. Zeissler, and E. Noth, "Age and Gender Recognition Based on Multiple Systems Early vs. Late fusion," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2830–2833, 2010.

[22] R. Porat, D. Lange, and Y. Zigel, "Age recognition based on speech signals using weights supervector," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2814–2817, 2010.

[23] M. Kockmann, L. Burget, and J. Cernocky, "Brno University of Technology System for Interspeech 2010," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2822–2825, 2010.

[24] P. Nguyen, T. Le, D. Tran, X. Huang, and D. Sharma, "Fuzzy Support Vector Machines for Age and Gender Classification," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2806–2809, 2010.

[25] M. Feld, E. Barnard, C. van Heerden, and C. Muller, "Multilingual speaker age recognition: Regression analyses on the Lwazi corpus," in *Automatic Speech Recognition and Understanding, 2009. ASRU 2009. IEEE Workshop on*, pp. 534–539, 2009.

[26] D. v. Leeuwen and M. Bahari, "Calibration of probabilistic age recognition," 2012.

[27] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[28] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[29] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, pp. 857–860, 2011.

[30] M. H. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," in *Proceedings ICASSP'2013*, pp. 7344–7348, 2013.

[31] M. H. Bahari, M. McLaren, H. Van hamme, and D. van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Interspeech*, pp. 506–509, 2012.

[32] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, vol. 4, 2006.

[33] C. Lu, T. Lee, and C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.

[34] A. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[35] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[36] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least squares support vector machines*. World Scientific, 2002.

[37] I. Fodor, "Statistical techniques to find similar objects in images," in *Proc. the American Statistical Association, Statistical Computing Section*, 2003.

[38] J. A. Suykens, L. Lukas, and J. Vandewalle, "Sparse approximation using least squares support vector machines," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 2, pp. 757–760, 2000.

[39] Y. Li, C. Lin, and W. Zhang, "Improved sparse least-squares support vector machine classifiers," *Neurocomputing*, vol. 69, no. 13, pp. 1655–1658, 2006.

[40] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, no. 5, pp. 980–988, 2008.

[41] M. McLaren and D. van Leeuwen, "Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 20, no. 3, pp. 755–766, 2012.

[42] M. McLaren and D. van Leeuwen, "A simple and effective speech activity detection algorithm for telephone and microphone speech," in *Proc. NIST SRE Workshop*, pp. 1–6, 2011.

[43] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," pp. 213–218, 2001.

[44] M. Mandasari, M. McLaren, and D. van Leeuwen, "Evaluation of i-vector speaker recognition systems for forensic application," in *Proc. Inter*, 2011.

[45] M. I. Mandasari, R. Saeidi, M. McLaren, and D. A. van Leeuwen, "Quality Measure Functions for Calibration of Speaker Recognition System in Various Duration Conditions," *IEEE Transactions on Acoustics, Speech, and Language Processing*, no. 99, 2013.

[46] A. Braun and L. Cerrato, "Estimating speaker age across languages," in *Proc. ICPhS*, vol. 99, pp. 1369–1372, 1999.

Chapter 3

Gender and age recognition using Gaussian weights

This chapter is based on the following articles:
1) Bahari, M.H., Van hamme, H. (2011), "Speaker age estimation and gender detection based on supervised non-negative matrix factorization," IEEE workshop biometric measurements and systems for security and medical applications, pp. 1-6, Italy.
2) Bahari, M.H., Van hamme, H. (2012), "Speaker Age Estimation using Hidden Markov model weight supervectors," 11th conf. information science, signal processing and their applications, pp. 517-521, Canada.

3.1 Abstract

This chapter proposes a new approach for speaker gender and age identification. In this method, utterances are modeled using hidden Markov model (HMM) weight supervectors. Then, a weighted supervised non-negative matrix factorization (WSNMF) is applied to reduce the dimension of the input space and recognize the age-gender category of the speaker. Finally, a least squares support vector machine regressor (LSSVR) is employed to estimate the age of speakers using the obtained low-dimensional vectors. Evaluation results on a corpus of read and spontaneous speech in Dutch confirm the effectiveness of the proposed scheme.

3.2 Introduction

Speech patterns reflect different characteristics of the speaker. For example, a person's speech pattern can provide information about his/her age, gender, dialect, emotional state and even membership of a particular social or regional group [1]. Profiling a person along different characteristics from his/her voice patterns is required in many commercial applications.

In this research, we focus on speaker age and gender. Since the perceptions of gender and age have a significant mutual impact on each other, these two characteristics are studied together in many publications [3-4].

Computerized speech-based age estimation is challenging from different points of view. First, usually there exists a difference between the age of a speaker as perceived, namely the perceptual age, and their actual age, namely the chronological age. Second, developing a robust age estimation method requires a labeled, wide age-range and balanced database. Third, voice patterns are affected by many parameters, such as smoking and gender, i.e. there is a significant intra-speaker variability that is not related to or only correlated with age.

The problem of age group recognition has been addressed previouosly [2,3,14,15]. For example, Bocklet and his colleagues introduced a method based on a Gaussian mixture model (GMM) mean supervector and a Support Vector Machine (SVM) to classify speakers into seven age-gender categories [2]. They used Mel Frequency Cepstral Coefficients (MFFCs) as features in their recognizer. Although this method was attractive from several aspects, it demands working with very large dimensions for a GMM with a large number of mixtures. In [3], the GMM universal background model is merged with the SVM classifier and the problem of high dimensional supervectors is tackled by using Gaussian

mixture weight supervectors, which have a lower dimension compared to mean or variance supervectors. In [14], Supervised Non-Negative Matrix Factorization (SNMF) was employed to classify speakers based on their age group and genders. In this method, HMM weight supervectors were applied instead of GMM weight or GMM mean supervectors. Zhang et. al. reported age and gender recognition results with the use of an unsupervised Non-negative Matrix Factorization (NMF) over Gaussian mixture weight supervectors in [18]. In their approach, the acoustic features consist of Mel Spectra with mean normalization and Vocal Tract Length Normalization (VTLN) [19], augmented with their first and second order time derivatives. Although their method could recognize the gender of speakers with high accuracy, it is not very successful for age estimation. They also conclude that adding VTLN decreases the accuracy of gender detection but it helps in age recognition.

In this chapter, a new gender detection and age estimation approach is introduced. To develop this method, we first determine an acoustic model for each speaker of the database by adapting the speaker independent model to the data of each utterance using the maximum likelihood (ML) re-estimation approach. Then, Gaussian mixture weights are extracted and concatenated to form a supervector for each utterance. Next, WSNMF is employed to reduce the dimension of the input space. Finally, the age of a speaker is estimated using a least squares support vector machine regressor (LSSVR) over the low-dimensional vectors obtained using WSNMF.

This chapter is organized as follows. Section 2 introduces WSNMF and LSSVR. In Section 3, the proposed approach is elaborated. The evaluation results are illustrated in Section 4. The chapter finishes with a conclusion in Section 5.

3.3 Background

In this section, the applied mathematical tools including WSNMF and LSSVR are briefly reviewed.

3.3.1 Weighted Supervised NMF

NMF is a popular machine learning algorithm [6], which is successfully applied to different sound and speech processing applications [7-8]. During the last decade, different extensions of NMF such as Supervised NMF (SNMF) [8] and Weighted Supervised NMF (WSNMF) [4] have been developed to solve real world problems. SNMF and WSNMF are originally developed for supervised pattern

recognition. However, they can also be applied as dimensionality reduction methods.

WSNMF for pattern recognition

The problem addressed by a WSNMF pattern recognizer is defined as follows. Assume that we are given a training dataset $S^{\text{tr}} = \{(\mathbf{w}_1, \mathbf{y}_1), \ldots, (\mathbf{w}_s, \mathbf{y}_s), \ldots, (\mathbf{w}_S, \mathbf{y}_S)\}$, where $\mathbf{w}_s$ denotes a vector of observed features of the data item and $\mathbf{y}_s$ denotes a label vector, i.e. a vector containing one in the d^{th} row if $\mathbf{w}_s$ belongs to the d^{th} class and zeros elsewhere. A vector can be a member of multiple classes, i.e. $\mathbf{y}_s$ can have multiple non-zero elements. The goal is to approximate a classifier function g, such that for an unseen observation $\mathbf{w}^{\text{tst}}$, $\hat{y} = g(\mathbf{w}^{\text{tst}})$ is as close as possible to the true label. If all elements of S^{tr} are non-negative, this problem can be solved by SWNMF directly. First, training data is used to form a matrix $\mathbf{W}^{\text{tr}}$ as follows:

$$\mathbf{W}^{\text{tr}}_{\text{up}} = [\mathbf{y}_1 \ldots \mathbf{y}_S] \tag{3.1}$$

$$\mathbf{W}^{\text{tr}}_{\text{dwn}} = [\mathbf{w}_1 \ldots \mathbf{w}_S] \tag{3.2}$$

$$\mathbf{W}^{\text{tr}} = \begin{bmatrix} \mathbf{W}^{\text{tr}}_{\text{up}} \\ \\ \mathbf{W}^{\text{tr}}_{\text{dwn}} \end{bmatrix} \tag{3.3}$$

Then, the non-negative matrix $\mathbf{W}^{\text{tr}}$, which is of size $M \times S$, is decomposed into two new non-negative matrices, namely $\mathbf{B}^{\text{tr}}$ and $\mathbf{H}^{\text{tr}}$ of size $M \times Z$ and $Z \times S$ respectively.

$$\begin{bmatrix} \mathbf{W}^{\text{tr}}_{\text{up}} \\ \\ \mathbf{W}^{\text{tr}}_{\text{dwn}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{B}^{\text{tr}}_{\text{up}} \\ \\ \mathbf{B}^{\text{tr}}_{\text{dwn}} \end{bmatrix} \mathbf{H}^{\text{tr}} \tag{3.4}$$

$$\mathbf{B}^{\text{tr}} = \begin{bmatrix} \mathbf{B}^{\text{tr}}_{\text{up}} \\ \\ \mathbf{B}^{\text{tr}}_{\text{dwn}} \end{bmatrix} \tag{3.5}$$

This factorization is performed by minimizing the following extended Kullback-Leibler divergence:

$$\Delta^{kl}\left(\mathbf{W}^{\text{tr}}, \mathbf{B}^{\text{tr}}\mathbf{H}^{\text{tr}}\right) =$$

$$\sum_{m,s} \mathbf{\Gamma}_{m,s} \left[\mathbf{W}^{\text{tr}}_{m,s} \log \left[\frac{\mathbf{W}^{\text{tr}}_{\text{m,s}}}{(\mathbf{B}^{\text{tr}}\mathbf{H}^{\text{tr}})_{\text{m,s}}} \right] (\mathbf{B}^{\text{tr}}\mathbf{H}^{\text{tr}})_{\text{m,s}} - \mathbf{W}^{\text{tr}}_{\text{m,s}} \right] + \rho \sum_{z,s} \mathbf{H}^{\text{tr}}_{z,s} \tag{3.6}$$

The last term penalizes large entries in $\mathbf{H}^{\text{tr}}$, so ρ controls the sparsity of $\mathbf{H}^{\text{tr}}$. It can be shown that the above-mentioned function is non-increasing under the following multiplicative updating rules [13]:

$$\mathbf{B}^{\text{tr}} \leftarrow \left\{ \frac{[\mathbf{B}^{\text{tr}}]}{[\mathbf{\Gamma}(\mathbf{H}^{\text{tr}})']} \right\} \circ \left\{ \frac{[\mathbf{\Gamma} \circ \mathbf{W}^{\text{tr}}]}{[\mathbf{B}^{\text{tr}}\mathbf{H}^{\text{tr}}]} \left(\mathbf{H}^{\text{tr}}\right)' \right\} \tag{3.7}$$

$$\mathbf{H}^{\text{tr}} \leftarrow \left\{ \frac{[\mathbf{H}^{\text{tr}}]}{[(\mathbf{B}^{\text{tr}})'\mathbf{\Gamma} + \rho]} \right\} \circ \left\{ \left(\mathbf{B}^{\text{tr}}\right)' \frac{[\mathbf{\Gamma} \circ \mathbf{W}^{\text{tr}}]}{[\mathbf{B}^{\text{tr}}\mathbf{H}^{\text{tr}}]} \right\}, \tag{3.8}$$

where $\mathbf{A} \circ \mathbf{B}$ and $\frac{[\mathbf{A}]}{[\mathbf{B}]}$ are the element-wise product and division of matrices $\mathbf{A}$ and $\mathbf{B}$ respectively, the sign $'$ is the transpose operator and $\mathbf{\Gamma}$ is a weighting matrix with the same size as $\mathbf{W}^{\text{tr}}$, which is determined as follows:

$$\mathbf{\Gamma}^{\text{tr}}_{\text{up}} = \beta \mathbf{1}_{D \times S} \tag{3.9}$$

$$\mathbf{\Gamma}^{\text{tr}}_{\text{dwn}} = \mathbf{1}_{P \times S} \tag{3.10}$$

$$\mathbf{\Gamma}^{\text{tr}} = \begin{bmatrix} \mathbf{\Gamma}^{\text{tr}}_{\text{up}} \\ \\ \mathbf{\Gamma}^{\text{tr}}_{\text{dwn}} \end{bmatrix} \tag{3.11}$$

where $\mathbf{1}_{D \times S}$ and $\mathbf{1}_{P \times S}$ ($M = D + P$) are two matrices with the same size as $\mathbf{W}^{\text{tr}}_{\text{up}}$ and $\mathbf{W}^{\text{tr}}_{\text{dwn}}$ respectively with all their elements equal to one. The parameter β is a factor determining the importance of the supervision information. A reasonable value for this factor, which is also used in this chapter, is

$$\beta = \frac{\sum_{p,s} \mathbf{W}^{\text{tr}}_{\text{dwn}}}{\sum_{d,s} \mathbf{W}^{\text{tr}}_{\text{up}}} \tag{3.12}$$

Calculation of $\mathbf{B}^{\text{tr}}$ by factorizing the $\mathbf{W}^{\text{tr}}$ is called training the WSNMF. In the testing phase, $\mathbf{B}^{\text{tr}}$ which was obtained from the training phase, is used to determine the class label of unseen patterns, $\mathbf{w}^{\text{tst}}$, as follows:

$$\mathbf{h}^{\text{tst}} = \underset{\mathbf{h}^{\text{tst}}}{\text{argmin}} \Delta^{kl} \left(\mathbf{w}^{\text{tst}}, \mathbf{B}^{\text{tr}}_{\text{dwn}} \mathbf{h}^{\text{tst}}\right) \tag{3.13}$$

$$\hat{\mathbf{y}} = g(\mathbf{w}^{\text{tst}}) = \mathbf{B}^{\text{tr}}_{\text{up}} \mathbf{h}^{\text{tst}} \tag{3.14}$$

Notice that $\hat{\mathbf{y}}$ returns a fuzzy class membership that requires a decision criterion, such as thresholding or selecting the maximum entry.

WSNMF for dimensionality reduction

In many real-world applications, the dimension of the input space is very large. To reduce this dimension, different methods are suggested such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and NMF [6,10]. In this chapter WSNMF is applied to reduce the dimension of the input space. In this approach, training is performed on the columns of $\mathbf{H}^{\text{tr}}$ instead of the columns of $\mathbf{W}^{\text{tr}}_{\text{dwn}}$ and testing is performed on $\mathbf{h}^{\text{tst}}$ instead of $\mathbf{w}^{\text{tst}}$. Hence, the dimension of the input space is reduced from M to Z in the training and testing phases. Notice that many other conventional dimensionality reduction approaches such as PCA, which suffer from high computational complexity, are useless for the current problem because of the large input dimensionality (M= 666192).

3.3.2 LSSVR

Least squares support vector machine (LSSVM), which is a variant of SVM, was introduced by Suykens and Vandewalle [9]. It is employed as a machine learning tool for regression, clustering and classification tasks. Compared to SVM, LSSVM enjoys a faster training process because the quadratic programming problem of SVM is reduced to that of solving a system of linear equations. Furthermore, the LSSVM formulation involves fewer tuning parameters [10].

A continuous function can be fitted to the training data with a least squares support vector regressor (LSSVR), a technique which shares many of the advantages of LSSVM classification [9]. In this research, LSSVR is applied to estimate the age of speakers.

3.4 Proposed Approach

In this section, the proposed approach for age estimation is elaborated. To introduce this method, first the procedure of forming a supervector for a speaker is explained. Then, the proposed scheme in the training and testing phases is elucidated in detail.

3.4.1 Feature selection, acoustic model and supervectors

The acoustic features consist of Mel log-spectra with mean normalization and vocal tract length normalization [5], augmented with their first and second order time derivatives. These features are then mapped to a 36 dimensional acoustic

space using mutual information discriminant analysis (MIDA) [11]. The acoustic model uses a shared pool of 49740 Gaussians to model the observations in 3873 cross-word context-dependent tied triphone HMM states, each modeled with a Gaussian mixture of Eq. 3.15. All acoustic units —context-dependent variants of one of the 46 phones, silence, garbage and speaker noise— have a 3-state left-to-right topology. The speaker independent acoustic model is trained on the CGN corpus (Corpus Gesproken Nederlands) [17].

The speaker dependent HMM weights for each speaker of the N-Best evaluation corpus [12], result from a maximum likelihood re-estimation of the speaker independent weights based on a forced alignment of the training data for that speaker using the speaker-independent acoustic model. Subsequently, the Gaussian mixture weights are extracted and concatenated to form a supervector for each speaker.

Consider the j^{th} state of speaker independent HMM Gaussian mixture with the following likelihood function of the data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_\tau\}$:

$$p(\mathbf{x}_t|\lambda^j) = \sum_{c=1}^{C} b_c^j p(\mathbf{x}_t|\mu_c^j, \mathbf{\Sigma}_c^j)$$

$$\lambda^j = \{b_c^j, \mu_c^j, \mathbf{\Sigma}_c^j\},\ c = 1, \ldots C, \tag{3.15}$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c^j is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c^j, \mathbf{\Sigma}_c^j)$ is a Gaussian probability density function with mean μ_c^j and covariance matrix $\mathbf{\Sigma}_c^j$, and C is the total number of Gaussians in the mixture.

Given an utterance, maximum likelihood re-estimation of the weight for the c^{th} mixture component is calculated as follows:

$$\omega_c^j = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{b_c^j p(\mathbf{x}_t|\mu_c^j, \mathbf{\Sigma}_c^j)}{\sum_{c=1}^{C} b_c^j p(\mathbf{x}_t|\mu_c^j, \mathbf{\Sigma}_c^j)} \tag{3.16}$$

where τ is the total number of frames in the utterance. Finally, the weight supervector of the given utterance is formed as follows.

$$\mathbf{w}^j = [\omega_1^j, \ldots, \omega_c^j, \ldots, \omega_C^j] \tag{3.17}$$

$$\mathbf{w} = [\mathbf{w}^1, \ldots, \mathbf{w}^j, \ldots, \mathbf{w}^J]' \tag{3.18}$$

3.4.2 Training Phase

The block diagram of the training phase of the proposed method is illustrated in Figure 3.1. As it can be interpreted from this figure, the speech signal

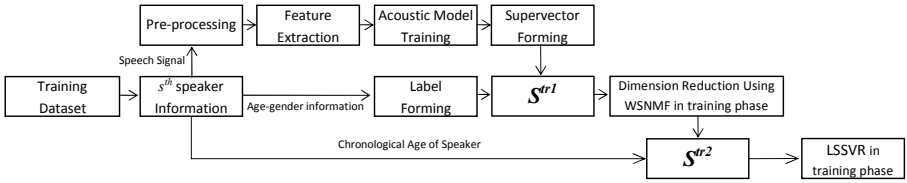

Figure 3.1: *The block-diagram of the proposed method in the training phase.*

of each speaker in the training dataset is used to form one supervector per speaker. Then, an age-gender category label is formed for each supervector. Each label is a vector with dimension equal to the total number of considered age-gender categories. The label of the s^{th} utterance, $\mathcal{X}_s$, which belongs to the d^{th} category, is formed such that the d^{th} element of the label vector is equal to 1 and the other elements are equal to zero. For example, if the an utterance of the training dataset belongs to the second category, its label vector is $\mathbf{y}_s = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}'$.

After calculating all S supervectors for all S utterances of the training data set and labeling them with their age-gender category, we obtain $S^{\text{tr1}} = \{(\mathbf{w}_1, \mathbf{y}_1), \ldots, (\mathbf{w}_s, \mathbf{y}_s), \ldots, (\mathbf{w}_S, \mathbf{y}_S)\}$. This dataset is utilized to reduce the dimension of the input space of the training dataset using the introduced WSNMF. In the factorization process, cost function (3.6) is minimized under the following constraint, which normalizes the columns of $\mathbf{B}^{\text{tr}}$ such that any linear combination yields valid mixture weights.

$$1 = \sum_{p \in Q_{j,z}} (\mathbf{B}^{\text{tr}})_{p,z} \text{ for all states } j \text{ and columns } z \tag{3.19}$$

where $Q_{j,z}$ is the set of elements of the z^{th} column of $\mathbf{B}^{\text{tr}}$, which correspond to the s^{th} state. The results of factorization are $\mathbf{B}^{\text{tr}}$, and $\mathbf{H}^{\text{tr}}$. Now the columns of $\mathbf{H}^{\text{tr}}$, which are of size Z, can be used instead of the columns of $\mathbf{W}^{\text{tr}}_{\text{dwn}}$, which are of size M. Consequently, a new input-output set can be formed using the columns of $\mathbf{H}^{\text{tr}}$ and the chronological age of their corresponding speaker (α^{tr}) so that $S^{\text{tr2}} = (\mathbf{H}^{\text{tr}}_1, \alpha^{\text{tr}}_1), \ldots, (\mathbf{H}^{\text{tr}}_s, \alpha^{\text{tr}}_s), \ldots, (\mathbf{H}^{\text{tr}}_S, \alpha^{\text{tr}}_S)$. This dataset is used to train the LSSVR with Gaussian kernel function. A 10-fold cross-validation approach is applied to tune the smoothing parameter of the kernels.

3.4.3 Testing Phase

Figure 3.2 indicates the architecture of proposed method in the testing phase. As can be interpreted from this figure, the procedure of obtaining the supervector

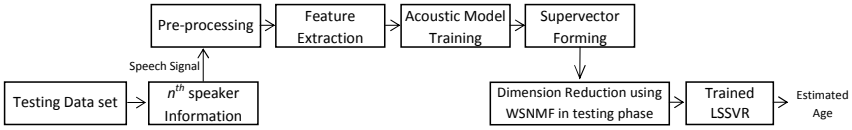

Figure 3.2: *The block-diagram of the proposed method in testing phases.*

of the GMM weights is repeated for each single speaker of the test dataset. Then, the trained WSNMF is applied to recognize the age-gender label of each supervector.

The block-diagram of the proposed method in the testing phase is demonstrated in Figure 3.2. A supervector per test speaker is formed with the same method as outlined in previous section. Then, the obtained supervector is fed into the trained WSNMF, where $\mathbf{B}^{\text{tr}}$ is fixed, to estimate $\mathbf{h}^{\text{tst}}$ of size Z using (3.13) and approximate age-gender label of speaker using (3.14). $\mathbf{h}^{\text{tst}}$, which is a low-dimension vector, is used as the input of the trained LSSVR. The output of the trained LSSVR is an estimation of the test speaker age.

3.5 Evaluation and Results

3.5.1 Corpus

Speech patterns of 425 speakers from the N-best evaluation corpus [12] were used. The corpus contains live and read commentaries, news, interviews, and reports broadcast in Belgium. Figure 3.3 and 3.4 show the age histogram of male and female speakers. To evaluate the proposed method, 5-fold cross-validation is used. Therefore, first all speakers in the database are divided into 5 disjoint folds. Then, five independent experiments are run so that in each experiment four folds are used as training dataset and the remaining fold is used as testing dataset.

3.5.2 Results

In all experiments, the sparsity parameter (ρ) is 1000.

The MAE of the proposed method for different values of Z is reported in Figure 3.5. For comparison purpose, the results of employing standard NMF [6] as a dimensionality reduction method instead of WSNMF are also included

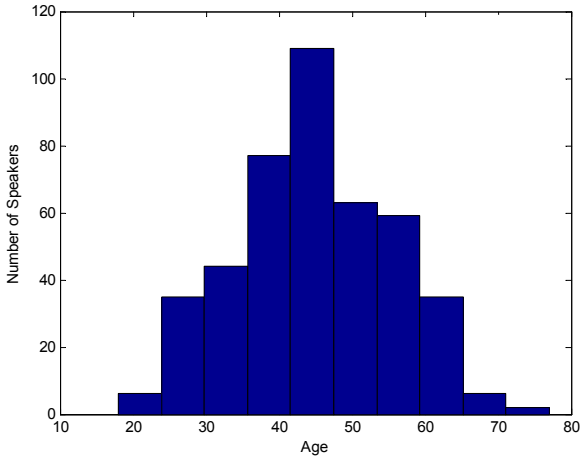

Figure 3.3: *Age histogram of male speakers in the evaluation corpus.*

in figure 3.5 under caption NMF. The MAE of our previously published age estimator [4], namely Hybrid WSNMF and GRNN (HWNN), is around 7.48 years. Therefore, the accuracy of the proposed method in speaker age estimation is better than the HWNN when the target dimension Z is between 20 and 70. The figure also suggests that using NMF is less effective than WSNMF for dimensionality reduction in this case.

Table 3.1 shows the average of age-group recognition accuracy over all performed experiments for the best obtained value of Z, which is 40. The second row lists the prior class probability, or "chance levels". Hence, the WSNMF method performs better than guessing. In this table, age range of young, middle and senior groups are 18-35, 36-45 and 46-80 years respectively.

The gender detection accuracy of the proposed method over all five experiments is 96%.

Table 3.1: *Age group recognition accuracy in %.*

Age Category	Young	Middle	Senior
Prior	25	36	39
Recognition Accuracy	38	40	65

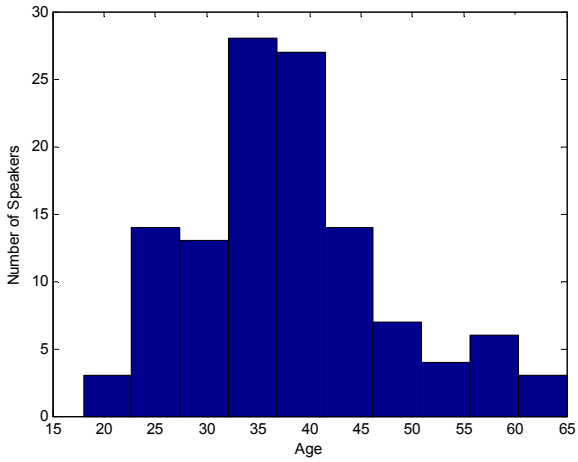

Figure 3.4: *Age histogram of female speakers in the evaluation corpus.*

3.6 Conclusions

In this chapter, a new hybrid method based on WSNMF and LSSVR has been proposed to identify speaker gender and age. In this method, WSNMF is applied to reduce the dimension of the input space, i.e. Gaussian weight supervectors, and a LSSVR is used to estimate the age of speakers. Evaluation on a Dutch database confirms the efficiency of the proposed method in speaker age estimation.

3.7 References

[1] D. C. Tanner, and M. E. Tanner, Forensic aspects of speech patterns: voice prints, speaker profiling, lie and intoxication detection. Lawyers & Judges Publishing, 2004.

[2] T. Bocklet, A. Maier, J. G. Bauer, F. Burkhardt, and E. Noth, "Age and gender recognition for telephone applications based on GMM supervectors and support vector machines, " In proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2008, pp. 1605-1608.

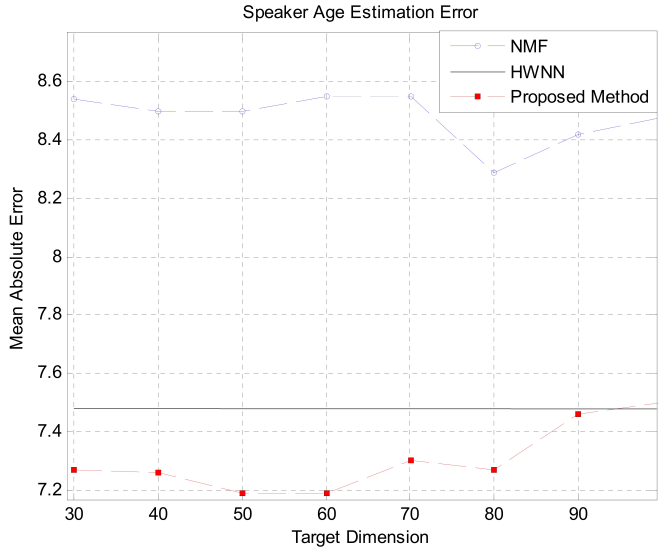

Figure 3.5: *The MAE of age estimation using the proposed method and NMF versus target dimension.*

[3] R. Porat, D. Lange, and Y. Zigel, "Age recognition based on speech signals using weights supervector, " In proc. Interspeech, 2010, pp. 2814-2817.

[4] M. H. Bahari, and H. Van hamme, "Speaker age estimation and gender recognition based on non-negative matrix factorization," In proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2011.

[5] J. Duchateau, M. Wigham, K. Demuynck, and H. Van hamme, "A flexible recogniser architecture in a reading tutor for children," ITRW on Speech Recognition and Intrinsic Variation, 2006, pp. 59-64.

[6] Lee, D. D., and H. S. Seung, "Algorithms for non-negative matrix factorization, " Advances in neural information processing systems, pp. 556-562, 2001.

[7] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria ," IEEE Trans. Audio, Speech, and Language Processing, vol. 15, no.3, pp. 1066-1074, 2007.

[8] H. Van hamme, “HAC-models: a novel approach to continuous speech recognition,” In proc. Interspeech, 2008, pp. 2554-2557.

[9] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines. World Scientific Pub. Co., Singapore, 2002.

[10] Fodor, I.K., “Statistical techniques to find similar objects in images,” Proceedings of the American Statistical Association, Statistical Computing Section, October 2003.

[11] K. Demuynck, Extracting, Modelling and Combining Information in Speech Recognition. Ph.D. thesis, Katholieke Universiteit Leuven, 2001.

[12] D. A. Van Leeuwen, J. Kessens, E. Sanders, and H. van den Heuvel, “Results of the n-best 2008 dutch speech recognition evaluation,” In proc. Interspeech, 2009, pp. 2571-2574.

[13] N. Ho, Nonnegative matrix factorization algorithms and applications. PhD thesis, Université. Catholique de Louvain, 2008.

[14] M. H. Bahari and H. Van hamme, “Age and Gender Recognition from Speech Patterns Based on Supervised Non-Negative Matrix Factorization,” In Proc. IAFPA 20th Annual Conference, 2011, pp 3-5.

[15] F. Metze, F. Metze, J. Ajmera, R. Englert, U. Bub, F. Burkhardt, J. Stegmann,C. Muller, R. Huber, B. Andrassy, J. Bauer, and B. Littel, ”Comparison of four approaches to age and gender recognition for telephone applications,” In proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), USA, pp. 1089-1092, 2007.

[16] G. Dobry, et. al. ”Supervector Dimension Reduction for Efficient Speaker Age Estimation Based on the Acoustic Speech Signal,” IEEE Trans. Audio, Speech, and Language Processing, vol. 19, no.v, pp. 1975-1985, 2011.

[17] N. H. J. Oostdijk and D. Broeder, “The spoken Dutch corpus and its exploitation environment“ in proc. 4th Int. Workshop on Linguistically Interpreted Corpora, Budapest, Hungary, 2003.

[18] X. Zhang, K. Demuynck, and H. Van hamme, “Rapid speaker adaptation with speaker adaptive training and non-negative matrix factorization,” In proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Czech republic, pp. 4456-4459, 2011.

[19] L.T Bosch, J. Driesen, H. Van hamme, and L. Boves, “On a computational model for language acquisition: modeling cross-speaker generalization,” In Proc. Int. Conf. Text, Speech and Dialogue, Czech Republic, pp. 315-322, 2009.

[19] A. Karsaz, H. Khaloozadeh, N. Pariz, and M. H. Bahari, "A new algorithm based on generalized target maneuver detection," In Proc. IEEE Int. Conf.Control and Automation (ICCA), pp. 3034-3039, 2007. [20] M.H. Bahari, H. Van hamme, "Rapid speaker adaptation using maximum likelihood neural regression," IEEE Int. Conf. multimedia and expo, Spain, pp. 1-6, 2011.

Chapter 4

Accent recognition using i-vectors, Gaussian weights and Gaussian means

This chapter is based on the following article:
1) Bahari, M.H., Saidi, R, Van hamme, H., van Leeuwen, D. (2013), "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," International conference on acoustics, speech, and signal processing (ICASSP), pp. 7344-7348, Canada.

4.1 Abstract

In this chapter, three utterance modelling approaches, namely Gaussian Mean Supervector (GMS), i-vector and Gaussian Weight Supervector (GWS), are applied to accent recognition problem. For each utterance modeling method, three different classifiers, namely the Support Vector Machine (SVM), the Naive Bayesian Classifier (NBC) and the Sparse Representation Classifier (SRC), are employed to find out suitable matches between the utterance modelling schemes and the classifiers. The evaluation database is formed by using English utterances of speakers whose native languages are Russian, Hindi, American English, Thai, Vietnamese and Cantonese. These utterances are drawn from the National Institute of Standards and Technology (NIST) 2008 Speaker Recognition Evaluation (SRE) database. The study results show that GWS and i-vector are more effective than GMS in this accent recognition task. It is also concluded that among the employed classifiers, the best matches for i-vector and GWS are SVM and SRC, respectively.

4.2 Introduction

A fundamental challenge of using Automatic Speech Recognition (ASR) systems in real world markets such as telephone networks and personal computers is their significant performance drop for non-native speakers [1, 2]. Consequently, accent/dialect recognition, has received an increased attention during the last years due to its importance for the enhancement of ASR performance [2]. It has also a wide range of commercial applications such as targeted advertising, service customization and forensics software. Although different methods have been suggested to solve this problem during the last decade, it still remains a challenging task.

Accent/dialect recognition techniques can be divided into phonotactic and acoustic approaches [3]. Since phonotactic features and acoustic (spectral and/or prosodic) features provide complementary cues, state-of-the-art methods usually apply a combination of both through a fusion of their output scores [3]. A phone recognizer followed by language models (PRLM) and parallel PRLM (PPRLM) techniques developed within the language recognition area, are successful phonotactic methods focusing on phone sequences as an important characteristic of different accents [4].

The acoustic approaches, which are the main focus of this chapter, enjoy the advantage of requiring no specialized language knowledge [3]. One effective acoustic method for accent recognition involves modeling speech recordings

with Gaussian mixture model (GMM) mean supervectors before using them as features in a support vector machine (SVM) [3]. Similar Gaussian mean supervector (GMS) techniques have been successfully applied to different speech analysis problems such as speaker recognition [5]. While effective, these features are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. Another effective approach for modeling the utterances is Gaussian weight supervector (GWS), which entails a lower dimension compared to GMSs [6, 7]. Recent studies show that the GWSs carry complementary information to GMSs [6, 8]. Consequently, incorporating them in the recognition system might increase the overall accuracy. A similar GWS framework was effectively applied to the problem of age and gender recognition [6, 9, 10]. In the field of speaker recognition, recent advances using i-vectors have increased the recognition accuracy considerably [11]. An i-vector is a compact representation of an utterance in the form of a low-dimensional feature vector. The same idea was also effectively applied to spoken language recognition and speaker age estimation [12, 13].

In this chapter, we apply GMSs, GWSs and i-vectors to recognize the native language of speakers from English spontaneous telephone speech recordings (L1 recognition problem).

To find out a suitable classifier for each modeling method, three different classifiers are tested, namely the Support Vector Machine (SVM), Naive Bayesian Classifier (NBC) and the Sparse Representation Classifier (SRC). The evaluation database is formed by using English utterances of speakers whose native languages are Russian, Hindi, American English, Thai, Vietnamese and Cantonese. These speech signals are extracted from the National Institute of Standards and Technology (NIST) 2008 Speaker Recognition Evaluation (SRE) corpus.

The rest of this chapter is organized as follows. Section 4.3 presents the related work and contributions of this chapter. In Section 4.4, the developed accent recognition systems are elaborated in details. Section 4.5 explains our experimental setup. The evaluation results are presented and discussed in section 4.6. The chapter ends with conclusions in section 4.7.

4.3 Related Work and Contributions

Different acoustic approaches developed in the area of language recognition have been suggested to reach a desirable accent recognition accuracy [3, 14, 15]. Recently Hanani *et al.* reported results of applying GMM-UBM, GMM-SVM (which is labeled as GMS-SVM in the rest of this chapter), and GMM

tokenization followed by n-gram language model methods to recognize 14 accents in the British Isles [3]. They used the Accents of the British Isles (ABI-1) corpus in their research. Their evaluation results show that GMS-SVM is more accurate compared to their other acoustic-based accent recognition systems.

DeMarco and Cox take this a step further by applying i-vectors to the same task [15]. They tested six different classification algorithms such as SVM and Linear Discriminant Analysis (LDA) and concluded that similar results as those of GMS-SVM can be obtained in the i-vector framework. Their results show no advantage for using i-vectors instead of GMSs.

In this chapter, we investigate the effectiveness of GMS and i-vector for accent recognition on a spontaneous and real speech database instead of the ABI-1 corpus, which consists of clean and read speech signals. Consequently, we formed a database of non-native accents of English by extracting English utterances with Russian, Hindi, American English, Thai, Vietnamese and Cantonese accents from the NIST 2008 SRE database. For each utterance modeling method, three different classifiers, namely SVM, NBC and SRC, are employed to further investigate the role of classifiers in this task. Unlike SVM and NBC, sparse representation classification techniques have never been tested on the accent recognition problem. On the other hand, recent studies show the proficiency of GWS in other speech technology problems such as speaker adaptation and speaker age group recognition [6, 8]. Consequently, we test GWS along with i-vectors and GMS in our investigations on accent recognition too.

4.4 System Description

4.4.1 Problem Formulation

In the accent or dialect recognition problem, we are given a training dataset $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \cdots, (\mathcal{X}_s, y_s), \cdots, (\mathcal{X}_S, y_S)\}$, where $\mathcal{X}_s$ denotes the s^{th} utterance of the training dataset and y_s denotes a label vector which shows the correct accent of the utterance. Each label vector contains a one in the d^{th} row if $\mathcal{X}_s$ belongs to the d^{th} class and zeros elsewhere. The goal is to approximate a classifier function (g), such that for an unseen observation $\mathcal{X}^{\text{tst}}$, $\hat{y} = g(\mathcal{X}^{\text{tst}})$ is as close as possible to the true label.

The first step for approximating the function g is converting variable-duration speech signals into fixed-dimensional vectors suitable for using in classification algorithms. Three approaches, namely GWS, GMS and i-vector are widely used for this purpose. These methods are described in section 4.4.2.

4.4.2 Utterance Modelling Approaches

In this section, the underlying ideas of GMS, GWS and i-vector are explained in more details.

Gaussian Weight Supervector

Consider a Universal Background Model (UBM) with the following likelihood function.

$$p(\mathbf{x}_t|\mu, \mathbf{\Sigma}) = \sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c) \tag{4.1}$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$ is a Gaussian probability density function with mean μ_c and covariance matrix $\mathbf{\Sigma}_c$ and C is the total number of Gaussians in the mixture (2048 in this work). Given an utterance, the occupancy posterior probability for the c^{th} mixture component is calculated as follows:

$$w_c = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)}{\sum_{j=1}^{C} b_j p(\mathbf{x}_t|\mu_j, \mathbf{\Sigma}_j)} \tag{4.2}$$

where τ is the total number of frames in the utterance. Finally, the GWS of the given utterance is formed as follows.

$$\mathbf{w} = [w_1, \cdots, w_c, \cdots, w_C] \tag{4.3}$$

Assuming the UBM components represent the acoustic space of all accents in the training dataset, each element in the GWS supervector of a sufficiently long utterance shows the existence level of the corresponding component in the utterance accent. This information can facilitate in the identification of accents.

Gaussian Mean Supervector

Given an utterance, a method such as Maximum-A-Posteriori adaptation is applied to adapt a Universal Background Model (UBM) to the speech characteristics of the speaker [5]. Then, the Gaussian means of the adapted GMM are extracted and concatenated to form a GMS for the given utterance.

i-vector

GMSs described in Section 4.4.2 have been shown to provide a good level of performance. In the related field of speaker recognition, GMSs are commonplace.

Recent progress in this field, however, has found an alternate method of modeling GMM supervectors that provides far superior speaker recognition performance [11]. This technique is referred to as total variability modeling. Total variability modeling assumes the GMM mean supervector, $\mathbf{m}$, that best represents a set of feature vectors can be decomposed as

$$\mathbf{m} = \mathbf{u} + \mathbf{Tv} \tag{4.4}$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace (400 dimensions in this work) and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean offset $\mathbf{Tv}$. The vector $\mathbf{v}$ is commonly referred to as the i-vector and has a standard normal distribution. The subspace matrix $\mathbf{T}$ is estimated via maximum likelihood in a large training dataset. An efficient procedure for training $\mathbf{T}$ and MAP adaptation of i-vectors $\mathbf{v}$ can be found in [16]. In the total variability modeling approach, i-vectors are the low-dimensional representation of an audio recording that can be used for classification and estimation purposes.

4.4.3 Classifiers

In this section, the applied classifiers are briefly described.

Naive Bayesian Classifier

Bayesian classifiers are probabilistic classifiers working based on Bayes' theorem and the maximum posteriori hypothesis. They predict class membership probabilities, i.e., the probability that a given test sample belongs to a particular class. The Naive Bayesian classifier (NBC) is a special case of Bayesian classifiers, which assumes class conditional independence to decrease the computational cost and training data requirement [17]. In this chapter, class distributions are assumed to be Gaussian.

Support Vector Machine

Support Vector Machines (SVM) is a supervised, binary and discriminative classifier initially introduced by Cortes and Vapnik [18]. Given a set of training examples, an SVM attempts to find the maximum margin separation hyperplane between two classes of data such that it generalizes well to the test data points. The basic SVMs are binary and discriminative classifiers, however, an effective multi-class and probabilistic extension has also been developed by Wu *et al.* based on pairwise coupling strategy [19].

Sparse Representation Classifier

Sparse representation classification techniques have received a great deal of attention in recent years. In sparse representation classification, first we search for a sparse representation of a test sample in terms of a linear combination of training samples. Then, the residuals for each class are calculated. These residuals show the level of similarity of the test sample with each category [20].

In our experiments, the dimension of feature vectors, i.e., the dimension of the GWS, GMS or i-vector, is greater than the number of training samples, which leads to an over-determined sparse representation problem. Therefore, to achieve the sparse representations of the test samples, we applied an l_1-minimization approach.

4.4.4 Training and Testing

The principle of the proposed accent recognition approach is illustrated in Figure 4.1. As it can be interpreted from this figure, in the training phase, each utterance in the train dataset is converted to a high dimensional vector using one of the three utterance modeling approaches (GWS, GMS or i-vector) described in Section 4.4.2. Then, the obtained high dimensional vector along with their corresponding accent label are used to train one of the three classifiers described in Section 4.4.3.

In the testing phase, the utterance modeling approach applied in the training phase is used to extract a high dimensional vector from the utterance of an unseen speaker. Then the trained classifier uses the extracted vector to recognize the accent of the test speaker.

4.5 Experimental Setup

4.5.1 Database

The National Institute for Standard in Technology (NIST) has held annual or biannual speaker recognition evaluations (SRE) for the past two decades. With each SRE, a large corpus of telephone (and more recently microphone) conversations are released along with an evaluation protocol. These conversations typically last 5 minutes and originate from a large number of participants for whom additional meta data is recorded—including participant age, language and smoking habits. The NIST databases where chosen for this work due to the large number of speakers and because the total variability

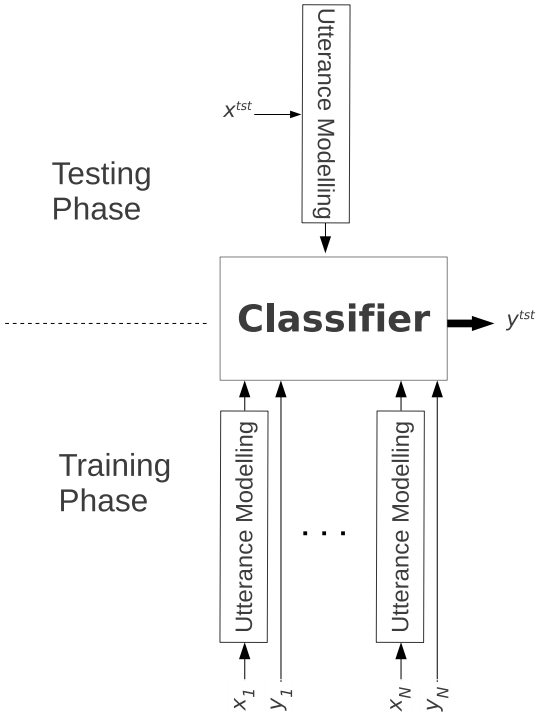

Figure 4.1: *The block diagram of the accent recognition systems in training and testing phases.*

subspace requires a considerable amount of development data for training. The development dataset used to train the total variability subspace and UBM includes over 30,000 speech recordings and was sourced from NIST 2004–2006 SRE databases, LDC releases of Switchboard 2 phase III and Switchboard Cellular (parts 1 and 2).

The NIST 2008 SRE database includes many English utterances from speakers whose native languages are Spanish, Russian, Hindi, etc. The native language of speakers usually affects their English pronunciation, i.e., accented speech, due to transferring the phonological rules from their native language into their English speech and creating innovative pronunciations for English sounds which do not exist in their mother tongue [21]. Unfortunately, the number of utterances in some accents is not high enough to perform our recognition experiments. Consequently, only five accents —Russian (RUS), Hindi (HIN), American English (USE), Thai (THA) and Vietnamese-Cantonese (VIE-YUH)— with

enough available recordings are chosen for our experiments. These utterances are extracted from telephone recordings of the core protocol, short2-short3, of the NIST 2008 SRE database. Note that since a fraction of Vietnamese Americans consists of Hoa people whose native language is Cantonese, Vietnamese and Cantonese are considered as one category in our experiments. Table 4.1 lists the number of utterances and speakers for each accent.

4.5.2 Performance Measure

The effectiveness of the proposed method is evaluated using the percentage of correctly classified utterances (P_{cc}) and minimum log-likelihood-ratio cost ($C_{\text{llr}}^{\text{min}}$) [22, 23]. This section briefly describes the applied performance measure methods.

Percentage of Correctly Classified Utterances

P_{cc} is a simple performance measure which can be calculated using the following relation.

$$P_{\text{cc}} = \frac{\kappa_{\text{cc}}}{\kappa} \times 100 \tag{4.5}$$

where κ_{cc} and κ denote the number of correctly classified utterances and the total number of utterances in the test dataset respectively.

Log-Likelihood Ratio Cost

Log-Likelihood Ratio Cost (C_{llr}) is an application-independent performance measure for recognizers with soft decisions output in the form of log-likelihood-ratios. This performance measure, which has been adopted for use in the NIST SRE, was initially developed for binary classification problems such as

Table 4.1: *The number of utterances and speakers for each accent category.*

Accent	Number of Utterances	Number of Speakers
USE	84	84
THA	63	41
RUS	49	32
HIN	62	39
VIE-YUH	101	69
Total	**359**	**265**

speaker recognition. It is extended to multi-class classification problems such as language recognition later in 2006 [22]. C_{llr} ranges between zero and infinity. For a perfect classifier without errors C_{llr} equals to zero, otherwise it is a positive number. The reference level of C_{llr} for indicating the effectiveness of classifier is $\log_2 D$, e.g. for a two class recognition problem reference level is $\log_2 D = 1$. For a useful recognizer, $C_{\text{llr}} < \log_2 D$ and for a poor input scores $C_{\text{llr}} > \log_2 D$, indicating that it would be better to apply prior information rather than the recognizer [22].

$C_{\text{llr}}^{\text{min}}$ represents the minimum possible C_{llr} which can be achieved for an optimally calibrated system.

4.6 Results

In this section, the performances of nine developed systems are evaluated and compared. The acoustic feature consists of 20 Mel-Frequency Cepstrum Coefficients (MFCCs) including energy appended with their first and second order derivatives, forming a 60 dimensional acoustic feature vector. This type of feature is very common in state-of-the-art i-vector based speaker recognition systems. To have more reliable features, Wiener filtering, speech activity detection [24] and feature warping have [25] been considered in front-end processing.

For the evaluation, a one speaker hold out training-testing strategy is adopted so that test speaker utterances are never included in the training set. In other words, 265 (total number of speakers in the database) independent experiments have been run. In each experiment, all utterances of a new speaker are used as testing and the rest of the utterances are used for training.

Table 4.2 lists the P_{cc} and $C_{\text{llr}}^{\text{min}}$ for all nine developed systems. For the SVM classifier different kernels have been tested and Table 4.2 shows only the best results obtained by the linear kernel. As it can be seen from Table 4.2, both classifier types and utterance modelling methods influence the recognition accuracy. While in SVM and NBC classification algorithms the i-vector framework leads to the most accurate recognition, for the SRC algorithm, GWS provides the best results.

The results also show that the NBC algorithm is not effective in this case. It can be due to high dimensionality of input features which increases class conditional dependency violating the naive assumption of the NBC (class conditional independence).

Table 4.2: *Comparison of various i-vector, GWS and GMS based systems. The results are given in* P_{cc} *and* $C_{\text{llr}}^{\text{min}}$.

Classifier	Feature	$P_{\text{cc}}(\%)$	$C_{\text{llr}}^{\text{min}}$
SVM	GMS	53	2.03
	GWS	58	1.92
	i-vector	56	1.77
NBC	GMS	47	2.12
	GWS	48	2.05
	i-vector	52	1.97
SRC	GMS	49	2.00
	GWS	56	1.63
	i-vector	41	2.08

Table 4.2 also illustrates that the GWS and the i-vector utterance modelling approaches are more effective than the GMS method in this non-native accents recognition task.

4.6.1 Feature Level Fusion

Many researches confirm the effectiveness of score level fusion [3, 26]. However, this type of fusion requires a development dataset which is not available in this task due to the limited number of utterances per accent. In this chapter, we employed feature level fusion requiring only one learning stage while taking advantage of mutual information [27]. In this type of fusion, the extracted i-vector, GWS and GMS of each utterance are concatenated to form a high dimensional supervector representing the utterance. Table 4.3 lists the results of NBC, SVM and SRC after feature level fusion. It shows that the accuracy of accent recognition increases after the fusion when SRC is applied for the classification. However, this improvement is not observed when NBC or SVM are employed.

Table 4.3: *Comparison of NBC, SVM and SRC after feature level fusion. The results are given in* P_{cc} *and* $C_{\text{llr}}^{\text{min}}$.

Classifier	Feature	$P_{\text{cc}}(\%)$	$C_{\text{llr}}^{\text{min}}$
NBC	i-vector-GWS-GMS	50	2.07
SVM	i-vector-GWS-GMS	56	1.84
SRC	i-vector-GWS-GMS	58	1.63

Table 4.4: *The confusion matrix of accent recognition for i-vector-GWS-GMS-SRC system. The results are given in percentage*

		Predicted				
		USE	THA	RUS	HIN	VIE-YUH
Actual	USE	65	4	7	6	18
	THA	14	46	2	3	35
	RUS	27	0	43	14	16
	HIN	8	5	3	60	24
	VIE-YUH	15	14	6	7	58

Table 4.4 illustrates the results of i-vector-GWS-GMS-SRC system as a confusion matrix. As it can be interpreted from this table, the recognition accuracy for all accents is noticeably higher than the chance level which confirms the efficiency of the proposed approach.

4.7 Conclusions

In this chapter, we have investigated the effectiveness of the GMS, GWS and i-vector utterance representation approaches for accent recognition on a spontaneous and real speech database formed by extracting English utterances with Russian, Hindi, American English, Thai, Vietnamese and Cantonese accents from the NIST 2008 SRE database. For each utterance modeling method, three different classifiers, namely SVM, NBC and SRC, have been employed to find out suitable matches between the utterance modelling schemes and the classifiers. The study results show that GWSs and i-vectors are more effective than GMS in this accent recognition task. Among the employed classifiers, the best matches for i-vector and GWS are SVM and SRC respectively. Furthermore, feature level fusion was found to be marginally effective in increasing the accent recognition accuracy, when SVM or SRC were applied as classifiers.

4.8 References

[1] A. Hanani, "Human and computer recognition of regional accents and ethnic groups from British English speech," *University of Birmingham*, July 2012.

[2] F. Biadsy, "Automatic Dialect and Accent Recognition and its Application to Speech Recognition," *Columbia University*, 2011.

[3] H. A., R. M.J., and C. M.J., "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech and Language*, vol. 27, no. 1, pp. 59–74, 2013.

[4] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[5] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[6] M. Li, K. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[7] X. Zhang, S. Hongbin, Z. Qingwei, and Y. Yonghong, "Using a kind of novel phonotactic information for SVM based speaker recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 92, no. 4, pp. 746–749, 2009.

[8] X. Zhang, K. Demuynck, and H. Van Hamme, "Latent variable speaker adaptation of Gaussian mixture weights and means," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4349–4352, 2012.

[9] M. Bahari and H. Van hamme, "Speaker age estimation and gender detection based on supervised Non-Negative Matrix Factorization," in *Proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pp. 1–6, 2011.

[10] M. Bahari and H. Van hamme, "Speaker age estimation using Hidden Markov Model weight supervectors," in *Proc. 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, 2012.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[12] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, pp. 857–860, 2011.

[13] M. Bahari, M. McLaren, H. Van hamme, and D. Van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Proc. Interspeech*, 2012.

[14] F. Biadsy, J. Hirschberg, and D. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," in *Proc. Interspeech*, 2011.

[15] A. Demarco and S. Cox, "Iterative Classification Of Regional British Accents In I-Vector Space," in *Proc. Machine Learning in Speech and Language Processing*, 2012.

[16] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, 2008.

[17] R. Yager, "An extension of the naive Bayesian classifier," *Information Sciences*, vol. 176, no. 5, pp. 577–588, 2006.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] T. Wu, C. Lin, and R. Weng, "Probability estimates for multi-class classification by pairwise coupling," *The Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[20] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[21] M. MacDonald, "The influence of Spanish phonology on the English spoken by United States Hispanics," *American Spanish pronunciation: Theoretical and applied perspectives*, 1989.

[22] N. Brummer and D. van Leeuwen, "On calibration of language recognition scores," in *Proc. IEEE Odyssey Speaker and Language Recognition Workshop*, pp. 1–8, 2006.

[23] N. Brummer, "Application-independent evaluation of speaker detection," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[24] M. McLaren and D. van Leeuwen, "A simple and effective speech activity detection algorithm for telephone and microphone speech," in *Proc. NIST SRE Workshop*, 2011.

[25] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," pp. 213–218, 2001.

[26] E. Wong and S. Sridharan, "Fusion of output scores on language identification system," pp. 1–11, 2003.

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

Chapter 5

Language and dialect recognition using non-negative factor analysis

This chapter is based on the following article:
Bahari, M.H., Dehak, N., Van hamme, H., Burget, L., Ali, A., Glass, J. (2014), "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," IEEE Transactions on Audio Speech and Language Processing (Accepted).

5.1 Abstract

Recent studies show that Gaussian mixture model (GMM) weights carry less, yet complementary, information to GMM means for language and dialect recognition. However, state-of-the-art language recognition systems usually do not use this information. In this research, a non-negative factor analysis (NFA) approach is developed for GMM weight decomposition and adaptation. This modeling, which is conceptually simple and computationally inexpensive, suggests a new low-dimensional utterance representation method using a factor analysis similar to that of the i-vector framework. The obtained subspace vectors are then applied in conjunction with i-vectors to the language/dialect recognition problem. The suggested approach is evaluated on the NIST 2011 and RATS language recognition evaluation (LRE) corpora and on the QCRI Arabic dialect recognition evaluation (DRE) corpus. The assessment results show that the proposed adaptation method yields more accurate recognition results compared to three conventional weight adaptation approaches, namely maximum likelihood re-estimation, non-negative matrix factorization, and a subspace multinomial model. Experimental results also show that the intermediate-level fusion of i-vectors and NFA subspace vectors improves the performance of the state-of-the-art i-vector framework especially for the case of short utterances.

5.2 Introduction

Language and dialect/accent recognition has received increased attention during the recent decades due to its importance for the enhancement of automatic speech recognition (ASR) [1, 2], multi-language translation systems, service customization, targeted advertising, and forensics softwares [3, 4].

Although research on text-independent language/dialect identification started in the early 1970s [5, 6], it remains a challenging task due to similarities of acoustic phonetics, phonotactics, and prosodic cues across different languages/dialects. Furthermore, in many practical cases we have no control over the available speech duration, channel characteristics, and noise level.

Recent language/dialect recognition techniques can be divided into phonotactic, and acoustic approaches [7]. Since phonotactic features and acoustic (spectral and/or prosodic) features provide complementary cues, state-of-the-art methods usually apply a combination of both through a fusion of their output scores [7]. A phone recognizer followed by language models (PRLM), parallel PRLM (PPRLM) and support vector machines PRLM techniques developed within the language recognition area, are successful phonotactic methods focusing on phone sequences as an important characteristic of different accents [8, 9].

The acoustic approaches, which are the main focus of this chapter, enjoy the advantage of requiring no specialized language knowledge [7]. One effective acoustic method for accent recognition involves modeling speech recordings with Gaussian mixture model (GMM) mean supervectors before using them as features in a support vector machine (SVM) [7]. Similar Gaussian mean supervector techniques have been successfully applied to different speech analysis problems such as speaker recognition [10]. While effective, these features are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. In the field of speaker recognition, recent advances using so-called i-vectors [11] have increased the classification accuracy considerably. The i-vector framework, which provides a compact representation of an utterance in the form of a low-dimensional feature vector, applies a simple factor analysis on GMM means. The same idea was also effectively applied in language/dialect recognition and speaker age estimation [12–14].

Recent studies show that GMM weights, which entail a lower dimension compared to Gaussian mean supervectors, carry less, yet complementary, information to GMM means [14–16]. Zhang *et al.* applied GMM weight adaptation in conjunction with mean adaptation for a large vocabulary speech recognition system to improve the word error rate [16]. Li *et al.* investigated the application of GMM weight supervectors in speaker age group recognition and showed that score-level fusion of classifiers based on GMM weights and GMM means improves recognition performance [15]. In [14] the feature level fusion of i-vectors, GMM mean supervectors, and GMM weight supervectors is applied to improve the accuracy of accent recognition.

Three main approaches have been suggested for GMM weights adaptation namely maximum likelihood re-estimation (ML) [17], non-negative matrix factorization (NMF) [16] and subspace multinomial model (SMM) [18]. The ML approach is conceptually simple and computationally inexpensive. However, the generalization of the adapted model is not guaranteed and only the observed weights are updated appropriately and the rest will be zero. This disadvantage affects the system performance especially for the case of short speech signals. The NMF expresses the adapted weights as a linear combination of a small number of latent vectors that are estimated on the training data [16]. This approach reduces the number of parameters that must be estimated from the enrollment data, and hence is more reliable in the context of short utterances. In this approach, the subspace matrix and the subspace vectors are assumed to be non-negative. This assumption makes the estimation of the subspace matrix more difficult. NMF is also very sensitive to initialization of the subspace matrix, which is often performed randomly. Inspired from the i-vector framework, Kockmann *et al.* introduced an approach for Gaussian weight supervector

decomposition for prosodic speaker verification [18]. The same approach was also used to apply intersession compensation in the context of phonotactic language recognition [19]. Soufifar *et al.* applied the same approach to extract low-dimensional phonotactic features for LRE [20, 21]. Although this method is attractive, it is computationally complex, and hence very time consuming.

In this research, we try to develop a new subspace method for GMM weight adaptation based on a factor analysis similar to that of the i-vector framework. In this method, namely non-negative factor analysis (NFA), the applied factor analysis is constrained such that the adapted GMM weights are non-negative and sum up to one. The proposed method is computationally simple and considerably faster than SMM. It also provides a wider bound for the adapted weights compared to that of the NMF. The obtained subspace vectors are applied to language and dialect recognition on three corpora, namely NIST 2011 LRE, QCRI Arabic DRE and RATS LRE. The GMM weight subspace vectors are fused with i-vectors effectively to form new vectors representing the utterances to improve the performance of the state-of-the-art i-vector framework for the language and dialect recognition tasks.

The rest of this chapter is organized as follows. Section 5.3 presents the background, and briefly describes the applied baseline systems. In Section 5.4, the proposed method is elaborated in detail. The evaluation results are presented and discussed in section 5.6. The chapter ends with conclusions in section 5.7.

5.3 Background

5.3.1 Problem Formulation

In the language/dialect recognition problem, we are given a training dataset $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \ldots, (\mathcal{X}_s, y_s), \ldots, (\mathcal{X}_S, y_S)\}$, where $\mathcal{X}_s$ denotes the s^{th} utterance of the training dataset, and y_s denotes a label vector that shows the correct language/dialect of the utterance. Each label vector contains a one in the d^{th} row if $\mathcal{X}_s$ belongs to the d^{th} class, and zeros elsewhere (the total number of categories is D). The goal is to approximate a classifier function (g), such that for an unseen observation $\mathcal{X}^{\text{tst}}$, $y = g(\mathcal{X}^{\text{tst}})$ is as close as possible to the true label.

The first step for approximating the function g is converting variable-duration speech signals into fixed-dimensional vectors suitable for classification algorithms. In this research, i-vectors, the GMM weight supervectors obtained by the ML method, the NMF subspace vectors, the SMM subspace vectors, and the NFA

subspace vectors are applied for this purpose, which are described in the following sections.

5.3.2 Universal Background Model

Consider a Universal Background Model (UBM) with the following likelihood function of data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_\tau\}$.

$$p(\mathbf{x}_t|\lambda) = \sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$$

$$\lambda = \{b_c, \mu_c, \mathbf{\Sigma}_c\},\ c = 1, \ldots C, \tag{5.1}$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$ is a Gaussian probability density function with mean μ_c and covariance matrix $\mathbf{\Sigma}_c$, C is the total number of Gaussians in the mixture, and τ is the total number of frames in the utterance. The parameters of the UBM –λ– are estimated on a large amount of training data representing different classes (languages/dialects).

5.3.3 i-vector Framework

One effective acoustic method for language/dialect recognition involves adapting UBM Gaussian means to the speech characteristics of the utterances. Then the Gaussian means of each adapted GMM are extracted and concatenated to form a supervector. Finally, the obtained Gaussian mean supervectors, which characterize the corresponding utterance, are applied to identify the language/dialect [2]. This method has been shown to provide a good level of performance in language/dialect recognition [2]. Recent progress in this field, however, has found an alternate method of modeling GMM mean supervectors that provides superior recognition performance [12]. This technique assumes the GMM mean supervector, $\mathbf{m}$, can be decomposed as

$$\mathbf{m} = \mathbf{u} + \mathbf{T}\mathbf{v}, \tag{5.2}$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean offset $\mathbf{T}\mathbf{v}$. The vector $\mathbf{v}$ is treated as a latent variable with the standard normal prior and the i-vector is its maximum-a-posteriori (MAP) point estimate. The subspace matrix $\mathbf{T}$ is estimated via maximum likelihood in a large training dataset. An efficient procedure for training $\mathbf{T}$ and for MAP adaptation of i-vectors can be found in [22]. In this approach, i-vectors are the low-dimensional

representation of an audio recording that can be used for classification and estimation purposes.

5.3.4 Conventional GMM Weight Adaptation Approaches

In this section, three main approaches of Gaussian weights adaptation are briefly described. In this chapter, the UBM weight and the adapted weight of the c^{th} Gaussian are denoted by b_c and w_c respectively.

Maximum Likelihood Re-estimation

In this method, the adapted weights w_c are obtained by maximizing the log-likelihood of Eq. 5.1 over the Gaussian weights. Rather than directly maximizing the log-likelihood of Eq. 5.1 we can also maximize the following auxiliary function over w_c

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c p(x_t|\mu_c, \mathbf{\Sigma}_c). \tag{5.3}$$

where $\gamma_{c,t}$ is the occupation count for the c^{th} mixture component and the t^{th} segment. Occupation counts are calculated as follows:

$$\gamma_{c,t} = \frac{b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)}{\sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)} \tag{5.4}$$

Maximizing Eq.5.3, will maximize the data likelihood [23].

Since $p(x_t|\mu_c, \mathbf{\Sigma}_c)$ remain unchanged in this maximization process, the auxiliary function Eq. 5.3 can be simplified to

$$\Phi(\lambda, w_c) = \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} \log w_c, \tag{5.5}$$

Finally, the adapted weights w_c after the first Expectation Maximization (EM) iteration are obtained as follows:

$$w_c = \frac{1}{\tau} \sum_{t=1}^{\tau} \gamma_{c,t} \tag{5.6}$$

Although maximum likelihood results are not yet reached after the first EM iteration, we will refer to this approach as ML re-estimation. In this chapter, neither in the ML re-estimation scheme nor in the weight adaptation methods given below, iterative re-insertion of the obtained adapted weights into $\gamma_{c,t}$ is used, i.e. the occupation counts $\gamma_{c,t}$ are obtained from the UBM and are kept fixed during the adaptation process.

Non-negative Matrix Factorization

The main assumption of the NMF based method [16] is that for a given utterance,

$$w_c = \mathbf{B}_c \mathbf{h}, \tag{5.7}$$

where $\mathbf{B}_c$ is a non-negative row vector forming the c^{th} row of the non-negative subspace matrix $\mathbf{B}$, and $\mathbf{h}$ is a low-dimensional and non-negative vector representing the utterance. In this method, $\mathbf{B}$ and $\mathbf{h}$ are initialized randomly, and then updated using the following multiplicative updating rules [24] to maximize the objective function Eq. 5.5.

$$\mathbf{B}_{c,\ell} \leftarrow \mathbf{B}_{c,\ell} \frac{\sum_s \mathbf{H}_{\ell,s} \bar{\gamma}_c(\mathcal{X}_s) / (\mathbf{BH})_{c,s}}{\sum_\imath \mathbf{H}_{\ell,\imath}} \tag{5.8}$$

$$\mathbf{H}_{\ell,s} \leftarrow \mathbf{H}_{\ell,s} \frac{\sum_c \mathbf{B}_{c,\ell} \bar{\gamma}_c(\mathcal{X}_s) / (\mathbf{BH})_{c,s}}{\sum_\jmath \mathbf{B}_{\jmath,\ell}}, \tag{5.9}$$

where $\bar{\gamma}_c(\mathcal{X}_s) = \sum_t \gamma_{c,t}(\mathcal{X}_s)$ and $\mathbf{H} = [\mathbf{h}_1 \dots \mathbf{h}_s \dots \mathbf{h}_S]$.

The adapted GMM weights are constrained to be non-negative and sum up to one. Since all elements of subspace matrix $\mathbf{B}$, and subspace vector $\mathbf{h}$ are non-negative, the adapted weights using NMF are also non-negative. To keep the sum of adapted GMM weights equal to one, the columns of subspace matrix $\mathbf{B}$ are normalized to sum up to one after updating it in each iteration. This normalization is also performed for the subspace vector $\mathbf{h}$. Details of this parameter re-estimation method can be found in [16].

The subspace matrix $\mathbf{B}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{h}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets can be used to classify languages/dialects.

Subspace Multinomial Model

Kockmann *et al.* introduced the SMM approach for Gaussian weight adaptation and decomposition with application to prosodic speaker verification [18]. The main assumption of this method is that for a given utterance,

$$w_c = \frac{\exp(z_c + \mathbf{A}_c \mathbf{q})}{\sum_{j=1}^{C} \exp(z_j + \mathbf{A}_j \mathbf{q})}, \tag{5.10}$$

where z_c is the c^{th} element of the origin of the supervector subspace, $\mathbf{A}_c$ is the c^{th} row of the subspace matrix and $\mathbf{q}$ is a low-dimensional vector representing the utterance.

In this method, $\mathbf{A}_c$ and $\mathbf{q}$ are estimated using a two-stage iterative algorithm similar to EM to maximize the objective function (5.5). For each stage of the EM-like algorithm, an iterative optimization approach similar to that of the Newton-Raphson scheme is applied as follows:

$$\mathbf{A}_c \leftarrow \mathbf{A}_c + \mathcal{H}_c^{-1}\Upsilon_c \tag{5.11}$$

$$\mathbf{q}_s \leftarrow \mathbf{q}_s + \mathcal{H}_s^{-1}\Upsilon_s, \tag{5.12}$$

where Υ_c and $\mathcal{H}_c$ are the gradient of the auxiliary function with respect to $\mathbf{A}_c$ and its corresponding approximated Hessian matrix respectively. Similarly Υ_s and $\mathcal{H}_s$ denote the gradient of the auxiliary function with respect to $\mathbf{q}_s$ and its corresponding approximated Hessian matrix respectively. Υ_c, $\mathcal{H}_c$, Υ_s and $\mathcal{H}_s$ are obtained as follows:

$$\Upsilon_c = \sum_{s=1}^{S}\left(\bar{\gamma}_c(\mathcal{X}_s) - w_{c,s}^{old}\sum_{c=1}^{C}\bar{\gamma}_c(\mathcal{X}_s)\right)\mathbf{q}_s^{'} \tag{5.13}$$

$$\mathcal{H}_c = \sum_{s=1}^{S}\max\left(\bar{\gamma}_c(\mathcal{X}_s), w_{c,s}^{old}\sum_{c=1}^{C}\bar{\gamma}_c(\mathcal{X}_s)\right)\mathbf{q}_s\mathbf{q}_s^{'} \tag{5.14}$$

$$\Upsilon_s = \sum_{c=1}^{C}\mathbf{A}_c^{'}\left(\bar{\gamma}_c(\mathcal{X}_s) - w_{c,s}^{old}\sum_{\imath=1}^{C}\bar{\gamma}_\imath(\mathcal{X}_s)\right) \tag{5.15}$$

$$\mathcal{H}_s = \sum_{c=1}^{C}\mathbf{A}_c^{'}\mathbf{A}_c\max\left(\bar{\gamma}_c(\mathcal{X}_s), w_{c,s}^{old}\sum_{\imath=1}^{C}\bar{\gamma}_\imath(\mathcal{X}_s)\right), \tag{5.16}$$

where $'$ denotes the symbol for transpose and $w_{c,s}^{old}$ is the c^{th} adapted weight for the s^{th} adaptation utterance obtained using the parameters from the preceding iteration.

Details of this parameter re-estimation approach, which involves calculation of the Hessian matrix and estimating the subspace vectors one-by-one, can be found in [18]. The subspace matrix $\mathbf{A}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{q}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets are used to classify languages/dialects.

5.4 Non-negative Factor Analysis

In this section, a new subspace method, namely Non-negative Factor Analysis (NFA), is introduced for GMM weight adaptation. The basic assumption of this

method is that for a given utterance, the c^{th} Gaussian weight of the adapted GMM (w_c) can be decomposed as follows

$$w_c = b_c + \mathbf{L}_c\mathbf{r}, \tag{5.17}$$

where b_c is the c^{th} weight of the UBM. $\mathbf{L}_c$ denotes the c^{th} row of the matrix $\mathbf{L}$, which is a matrix of dimension $C \times \varkappa$ spanning a low-dimensional subspace ($\varkappa \ll C$); $\mathbf{r}$ is a $\varkappa$-dimensional vector that best describes the utterance-dependent weight offset $\mathbf{Lr}$.

In this framework, neither subspace matrix $\mathbf{L}$ nor subspace vector $\mathbf{r}$ are constrained to be non-negative. However, unlike the i-vector framework, the applied factor analysis for estimating the subspace matrix $\mathbf{L}$ and the subspace vector $\mathbf{r}$ is constrained such that the adapted GMM weights are non-negative and sum up to one. The procedure of calculating $\mathbf{L}$ and $\mathbf{r}$ involves a two-stage algorithm similar to EM to maximize the objective function (5.5). In the first stage, $\mathbf{L}$ is assumed to be known, and we try to update $\mathbf{r}$. Similarly in the second stage, $\mathbf{r}$ is assumed to be known and we try to update $\mathbf{L}$. Each step is elaborated in the next subsections.

The subspace matrix $\mathbf{L}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{r}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets are used to classify languages and dialects in this chapter.

5.4.1 Updating Subspace Vector r

In the first stage of the applied iterative optimization procedure, vector $\mathbf{r}$ is estimated as follows:

Constrained optimization problem

Substituting w_c by $b_c + \mathbf{L}_c\mathbf{r}$ in the objective function of Eq. 5.5, we obtain

$$\Phi(\lambda, \mathbf{r}) = \sum_{t=1}^{\tau}\sum_{c=1}^{C} \gamma_{c,t} \log\left(b_c + \mathbf{L}_c\mathbf{r}\right) \tag{5.18}$$

or

$$\Phi(\lambda, \mathbf{r}) = \bar{\gamma}'(\mathcal{X}) \log\left(\mathbf{b} + \mathbf{Lr}\right), \tag{5.19}$$

where the log operates element-wise. $\mathbf{b}$ and $\bar{\gamma}(\mathcal{X})$ are obtained as follows,

$$\bar{\gamma}(\mathcal{X}) = \sum_{t} \begin{bmatrix}\gamma_{1,t} & \dots & \gamma_{C,t}\end{bmatrix}' \tag{5.20}$$

$$\mathbf{b} = \begin{bmatrix} b_1 & \dots & b_C \end{bmatrix}' \tag{5.21}$$

Given an utterance $\mathcal{X}$, a maximum likelihood estimation of $\mathbf{r}$ can be found by solving the following constrained optimization problem:

$$\max_{\mathbf{r}} \Phi(\lambda, \mathbf{r}) \tag{5.22}$$

Subject to

$\mathbf{1}(\mathbf{b} + \mathbf{L}\mathbf{r}) = 1$ Equality constraint

$\mathbf{b} + \mathbf{L}\mathbf{r} > 0$ Inequality constraint,

where $\mathbf{1}$ is a row vector of dimension C with all elements equal to 1. This constrained optimization problem has the following analytical solution for a square full-rank $\mathbf{L}$ (the proof for this relation is given in Appendix I):

$$\mathbf{r}(\mathcal{X}) = \mathbf{L}^{-1}\left[\frac{1}{\tau}\bar{\gamma}(\mathcal{X}) - \mathbf{b}\right] \tag{5.23}$$

For a skinny $\mathbf{L}$, where the number of rows is greater than the number of columns, solving this constrained optimization problem involves using iterative optimization approaches. Solving a constrained optimization problem is usually more time-consuming compared to an unconstrained one. Therefore, we relax the constraints, and convert the problem to an unconstrained optimization by the following simple tricks.

Reformulation of the equality constraint

The equality constraint is

$$\mathbf{1}\mathbf{b} + \mathbf{1}\mathbf{L}\mathbf{r} = 1. \tag{5.24}$$

We know that the UBM weights sum up to one, or $\mathbf{1}b = 1$. Hence

$$\mathbf{1}\mathbf{L}\mathbf{r} = 0. \tag{5.25}$$

If $\mathbf{1}$ is orthogonal to all columns of $\mathbf{L}$, i.e., $\mathbf{1}\mathbf{L} = 0$, the constraint of Eq. 5.25 holds for any possible $\mathbf{r}$. In the second stage of optimization, $\mathbf{L}$ is calculated such that $\mathbf{1}\mathbf{L} = 0$ holds.

Relaxing the inequality constraint

As can be seen in Eq. 5.22 there are C inequality constraints. If any inequality constraints are violated, the cost function of Eq. 5.22 cannot be evaluated.

In numerical optimization, if we start from a feasible point, there will be a "wall" over which we cannot climb, as the cost function becomes infinite at the boundary. Therefore, by controlling the steps of the maximization approach, violating the inequality constraint can be easily avoided. The exception is when any component of $\bar{\gamma}'(\mathcal{X})$ is zero. To avoid this problem, we replace zero elements of $\bar{\gamma}'(\mathcal{X})$ by very small positive values.

Maximization using gradient ascent

By simplifying the problem to an unconstrained maximization, different optimization techniques can be applied to obtain the maximum likelihood estimate of $\mathbf{r}$ in a reasonable time. We use a simple gradient ascent method with the following updating formula,

$$\mathbf{r} \leftarrow \mathbf{r} + \alpha_E \bigtriangledown \Phi(\lambda, \mathbf{r}) \tag{5.26}$$

$$\bigtriangledown \Phi(\lambda, \mathbf{r}) = \mathbf{L}' \frac{[\bar{\gamma}'(\mathcal{X})]}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})]}, \tag{5.27}$$

where $\frac{[.]}{[.]}$ denotes the element-wise division, α_E is the learning rate and $\bigtriangledown$ denotes gradient operator. In the first step of this method, α_E is set to a non-critical (non-negative) value and then it is reduced at each unsuccessful step (e.g. halved) and increased in each successful step (multiplied by 1.5). An unsuccessful iteration is when $\Phi(\lambda, \mathbf{r})$ decreases or any of the inequality constraints are violated. On our data, six successful gradient ascent iterations were enough for convergence of subspace vectors $\mathbf{r}$.

Initialization

Like many optimization problems, a bad initialization leads to a bad result. In this section, we try to obtain a reasonable initial point to be used in the iterative optimization algorithm. As mentioned, the constrained optimization problem has an analytical solution in the case of a square full-rank $\mathbf{L}$ given in Eq. 5.23. After reformulation explained in Section 5.4.1, $\mathbf{L}$ is never of full-rank. However, for a skinny $\mathbf{L}$, we can use the Moore-Penrose pseudo-inverse instead of the inverse to obtain a vector of the same dimension as $\mathbf{r}$.

$$\mathbf{r}_{pinv} = \mathbf{L}^{\dagger} \left[\frac{1}{\tau} \bar{\gamma}(\mathcal{X}) - \mathbf{b} \right] \tag{5.28}$$

where $\dagger$ is the sign for Moore-Penrose pseudo-inverse; $\mathbf{r}_{pinv}$ is an optimal solution for minimizing the Euclidean distance between $\frac{1}{\tau}\bar{\gamma}$ and $\mathbf{b} + \mathbf{L}\mathbf{r}$. However, this

solution ($\mathbf{r}_{pinv}$) may violate the inequality constraints of the problem, and hence be unfeasible. Since $w_c = b_c + \mathbf{L}_c\mathbf{r}$ and b_c are non-negative, a $\mathbf{r}$ with sufficiently small elements satisfies the inequality constraints. Therefore, by multiplying a small value θ to $\mathbf{r}_{pinv}$, we obtain a feasible initial point as follows:

$$\mathbf{r}_0 = \theta\mathbf{r}_{pinv} \tag{5.29}$$

We start from $\theta = 1$ and reduce (half) it until reaching a feasible initial point. On our data, $\theta = 0.1$ has been found small enough to obtain a feasible initial point.

5.4.2 Updating Subspace Matrix L

In the M-step, assuming $\mathbf{r}$ is known for all utterances in the training database, matrix $\mathbf{L}$ can be obtained by solving the following constrained optimization problem.

$$\max_{\mathbf{L}} \widetilde{\Phi}(\lambda, \mathbf{L}) \tag{5.30}$$

Subject to

$\mathbf{1}(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)) = 1$ Equality constraint

$\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s) > 0$ Inequality constraint

$s = 1, \ldots S,$

where

$$\widetilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \bar{\gamma}'(\mathcal{X}_s) \log\left[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)\right] \tag{5.31}$$

This constrained optimization problem has no analytical solution. Therefore, iterative optimization approaches are required.

As mentioned in Section 5.4.1, violating the inequality constraints can be avoided easily in numerical optimization by starting from a feasible initial point and controlling the step size.

All equality constraints can be simplified to a single constraint $\mathbf{1}\mathbf{L} = 0$ using the same trick mentioned in Section 5.4.1. To solve the resulting optimization problem with equality constraint $\mathbf{1}\mathbf{L} = 0$, a projected gradient algorithm [25] is

applied.

$$\mathbf{L}_i \leftarrow \mathbf{L} + \alpha_M \mathcal{P} \bigtriangledown \widetilde{\Phi}(\lambda, \mathbf{L}) \tag{5.32}$$

$$\bigtriangledown \widetilde{\Phi}(\lambda, \mathbf{L}) = \sum_s \frac{[\bar{\gamma}(\mathcal{X}_s)]}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X}_s)]} \mathbf{r}'(\mathcal{X}_s) \tag{5.33}$$

$$\mathcal{P} = \mathbf{I} - \frac{1}{C}\mathbf{1}'\mathbf{1}, \tag{5.34}$$

where α_M is the learning rate, $\mathbf{I}$ is an identity matrix of size C, and $\mathcal{P}$ is a projection also called the centering matrix. In the first step of this algorithm, α_M is set to a non-critical (non-negative) value and then it is reduced at each unsuccessful step (halved) and increased in each successful step (multiplied by 1.5). An unsuccessful iteration is when $\widetilde{\Phi}(\lambda, \mathbf{L})$ decreases, or any of the inequality constraints are violated. On our data, six successful gradient ascent iterations were enough for convergence of subspace matrix $\mathbf{L}$.

Initialization

We use Principal Component Analysis (PCA) for initialization of $\mathbf{L}$. In other words, we first form matrix $\mathbf{N}$ from the ML estimations of GMM weights for all training utterances as follows:

$$\mathbf{N} = \left[\frac{\bar{\gamma}(\mathcal{X}_1)}{\tau(1)}, \ldots, \frac{\bar{\gamma}(\mathcal{X}_s)}{\tau(s)}, \ldots, \frac{\bar{\gamma}(\mathcal{X}_S)}{\tau(S)} \right] \tag{5.35}$$

Then, the first $\varkappa$ principal components of $\mathbf{N}$ with high eigenvalues are used as initial point of $\mathbf{L}$ for maximization of $\widetilde{\Phi}(\lambda, \mathbf{L})$.

5.5 Comparison between NMF, SMM and NFA

In this section, flexibility and computational cost of NMF, SMM, and NFA are compared.

5.5.1 Modeling

Figure 5.1 shows the adapted weights of the UBM with three Gaussians using the ML re-estimation approach described in Section 5.3.4. In this figure, each dot shows the adapted weights using the ML approach for an utterance. Since the GMM weights are constrained to be positive, and sum up to 1, they are

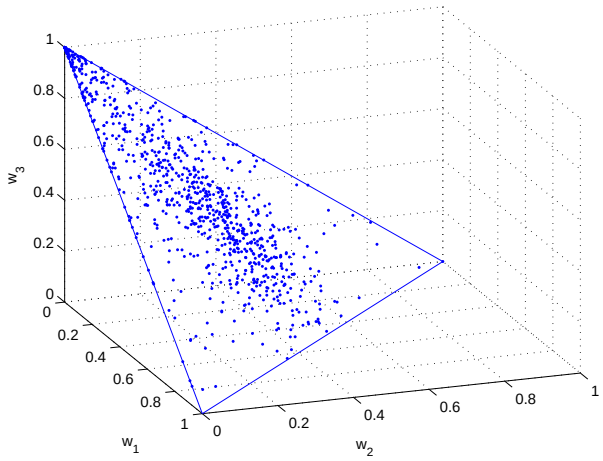

Figure 5.1: *The adapted weights of the UBM with three Gaussians using the ML method.*

embedded in a simplex. As shown in this figure, the adapted weights using the ML method can be very small—zero or very near zero— because the adapted weights of unobserved Gaussians or weakly observed Gaussians are zero or very near zero respectively. Consider the utterances and the UBM of Figure 5.1. Given these utterances as the training dataset, NMF, SMM and NFA are used to estimate a subspace matrices $\mathbf{B}$, $\mathbf{T}$ and $\mathbf{L}$ respectively.

For NMF, the straight line in Figure 5.2 shows the set of any possible adapted weights obtained using the estimated subspace matrix $\mathbf{B}$, which is of dimension 3×2 and was estimated after 300 iterations of the multiplicative updating algorithm [24] starting from a random initialization. Since $\mathbf{h}$ is non-negative and is normalized such that its elements sum up to one, the adapted weights using Eq. 5.7 make a convex combination of the columns of $\mathbf{B}$. Hence, the adapted weights are constrained to a bounded straight line on the simplex, as shown in Figure 5.2. As can be seen in this figure, although there are some data points near the border of the simplex, the straight line does not hit the border of the simplex. This shows that the subspace matrix $\mathbf{B}$ was not estimated appropriately. A closer analysis shows that this effect can be attributed to both slow convergence and falling into local minima. Depending on the initial value of $\mathbf{B}$, NMF may converge to an appropriate subspace matrix and the straight line can hit the border of the simplex. The multiplicative updating algorithm [24] does not guarantee convergence to the global minimum and is very sensitive to initialization, which is performed randomly in this example.

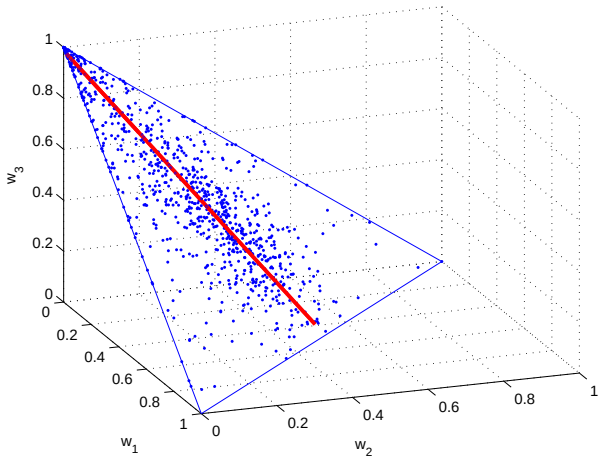

Figure 5.2: *The space of possible adapted weights of a UBM with three Gaussians using NMF.*

In the GMM weight adaptation problem, where the dimension of input data and the number of training datapoints are considerably greater than those of this example, this problem is expected to be even more challenging.

For the SMM, the curved line in Figure 5.3 shows the set of any possible adapted weights obtained using the estimated subspace matrix $\mathbf{A}$, which is of dimension 3×1. Since $\mathbf{q}$ is of dimension 1, and is not bounded, the adapted weights using Eq. 5.10 are embedded in a curved line hitting the corners of the simplex as shown in Figure 5.3. Since this curved line necessarily hits two corners of the simplex, the adapted weights can take on very small values for unobserved, or weakly observed, Gaussians in two dimensions as for the ML results. This problem is addressed in [26] by adding a regularization term. However, the regularization parameter requires fine-tuning over a development dataset [26].

For NFA, the straight line in Figure 5.4 shows the set of possible adapted weights obtained using the estimated subspace matrix $\mathbf{L}$, which is of dimension 3×1. Since $\mathbf{r}$ is of dimension 1, and is not constrained to be non-negative, the adapted weights using Eq. 5.17 are embedded in a straight line hitting the boundaries of the simplex as shown in Figure 5.4. This straight line does not necessarily hit the corners of the simplex*. This natural constraint makes it less flexible

*It nearly hits one corner of the simplex due to specific distribution of the given data in this example. However, this straight line, generally starts from a boundary of the simplex and ends at another boundary of it depending on the distribution of the data.

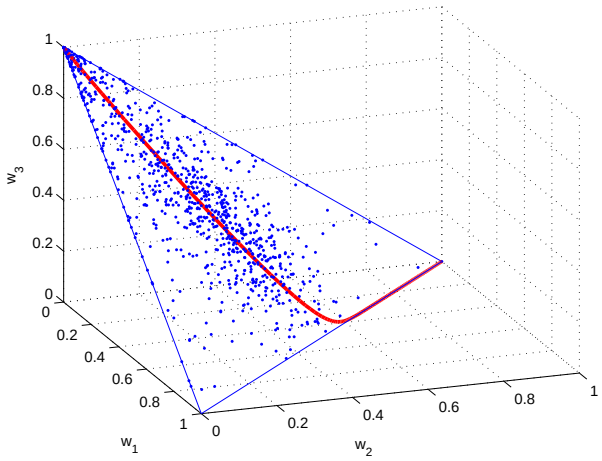

Figure 5.3: *The space of possible adapted weights of a UBM with three Gaussians using SMM.*

compared to SMM, where the adapted weights can take very small values due the the constraint that some simplex corner points are necessarily included in the obtained subspace. In contrast, both NMF and NFA avoid this problem because obtained subspaces of these approaches do not necessarily include simplex corners. The main difficulties of obtaining an appropriate subspace matrix in NMF are slow convergence rate, local optima and initialization, which will be further discussed in the next section.

5.5.2 Computation and Initialization

The procedure of updating the subspace matrix, and the subspace vectors is different between the NMF, SMM and NFA frameworks.

In the applied NMF, the subspace matrix and subspace vectors are randomly initialized, and then multiplicative updating rules are applied to update the subspace matrix and subspace vectors. As can be interpreted from Eq. 5.9, in NMF, the computational complexity of updating subspace vector $\mathbf{h}$ grows linearly with the subspace dimension. On our data, convergence was obtained in around 300 iterations.

In SMM, the initialization of the subspace matrix is similar to that of NFA, and the initial value of the subspace vectors is considered to be zero. SMM

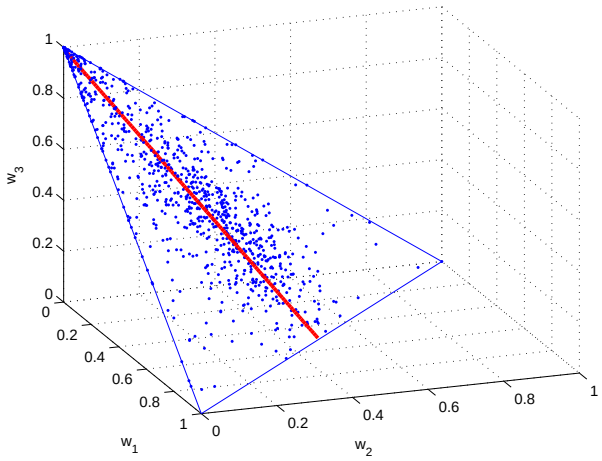

Figure 5.4: *The space of possible adapted weights of a UBM with three Gaussians using NFA.*

applies an optimization technique similar to that of Newton-Raphson, where the computational complexity of construction and inversion of the approximated Hessian matrix grows cubically with the subspace dimension. In this procedure, the subspace vectors are estimated one-by-one, which does not allow compilers to optimally exploit the parallelism of modern computer architectures, while matrix formulations as in NMF and NFA, do. On our data, convergence of SMM subspace matrix re-estimation was obtained in 10 iterations.

In NFA, the subspace matrix and subspace vectors are initialized as described in Sections 5.4.2 and 5.4.1, respectively. NFA applies a simple gradient ascent technique to estimate a subspace matrix and subspace vectors. As can be interpreted from Eq. 5.26, the computational complexity in each iteration of updating the subspace vector grows linearly with the subspace dimension. Like in NMF, in this technique, the corresponding subspace vectors for all utterances are treated as a single matrix, and then the gradient ascent technique is applied over the matrix. This makes the optimization significantly faster compared to estimating subspace vector for each utterance one-by-one. In this approach, convergence can be obtained in around 10 iterations of the applied two-stage optimization procedure.

Two-stage optimization approaches in NMF, SMM and NFA do not guarantee the convergence to the global minimum, and hence the initialization of the subspace matrices and the subspace vectors are critical. An important advantage

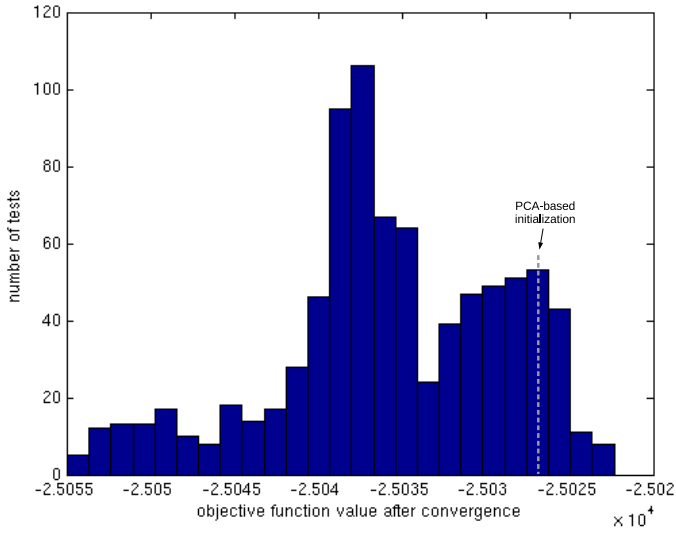

Figure 5.5: *The histogram of objective function value after convergence for 100 randomly initialized NFA factorizations.*

of SMM and NFA compared to NMF is that the subspace matrices of these methods are not constrained to be non-negative and PCA is used for their initialization as described in Section 5.4.2, while the initialization of the subspace matrix in NMF is more challenging as it is constrained to be non-negative.

To investigate the effect of the applied initialization in NFA, the toy problem of Section 5.5.1 is considered. Figure 5.5 shows the histogram of objective function value of the converged terials for over 850 randomly initialized NFA factorizations (subspace matrix initialization by random non-negative values is often used in NMF). The objective function value after convergence using the suggested initialization, which is shown by a dashed-line in the figure, is greater than that of NFA with random initialization in most of trials. Therefore, the suggested methods in Sections 5.4.2 and 5.4.1 yield a reasonable initial subspace matrix and subspace vectors to be used in the iterative optimization algorithm.

5.6 Experiments and Results

In this section, the performance of the proposed method and its characteristics are investigated on the NIST 2011 LRE, QCRI Arabic DRE and RATS LRE

corpora.

5.6.1 NIST 2011 LRE

Database

The National Institute of Science and Technology (NIST) 2011 LRE corpus is composed of 24 languages —Bengali, Dari, English-American, English-Indian, Farsi/Persian, Hindi, Mandarin, Pashto, Russian, Spanish, Tamil, Thai, Turkish, Ukrainian, Urdu, Arabic-Iraqi, Arabic-Levantine, Arabic-Maghrebi, Arabic-MSA, Czech, Lao, Punjabi, Polish, and Slovak— collected over telephone conversations and narrowband recordings. This evaluation set composed by three conditions based on the duration of the test segments. These durations are 30s, 10s and 3s.
The applied data for training and tuning are similar to that of the MIT Lincoln Laboratory (MITLL) system [27] submitted to the NIST 2011 LRE and were collected from the following sources:
• Telephone data from previous NIST (1996, 2003, 2005, 2007, 2009) LRE datasets, CallFriend, CallHome, Mixer , OHSU, and OGI-22 collections.
• Narrowband recordings collected from VOA broadcasts, Radio Free Asia, Radio Free Europe, and GALE broadcasts.
• Arabic corpora from LDC and Appen data were also obtained from telephone conversations, and some interview data.
• Some extra data were also obtained from Special Broadcast Services (SBS) in Australia.
• NIST 2011 LRE development data also included telephone conversations and narrowband broadcast segments.

UBM and Features

In this experiment, the applied UBM has 2048 mixtures, and acoustic features are exactly the same as that of the MIT Lincoln Laboratory (MITLL) NIST 2011 LRE submission [27]. They are based on cepstral features extracted using a sliding window of 20ms length, and 10ms overlap. These features were subjected to vocal tract length normalization followed by RASTA filtering [28]. The obtained cepstral features were converted to a Shifted Delta Cepstral (SDC) representation based on the 7-1-3-7 configuration [29]. This configuration produces a sequence of vectors of dimension 56. After extracting the SDC features and removing the non-speech frames, the feature vectors are mean and variance normalized over each speech recording. An intersession compensation

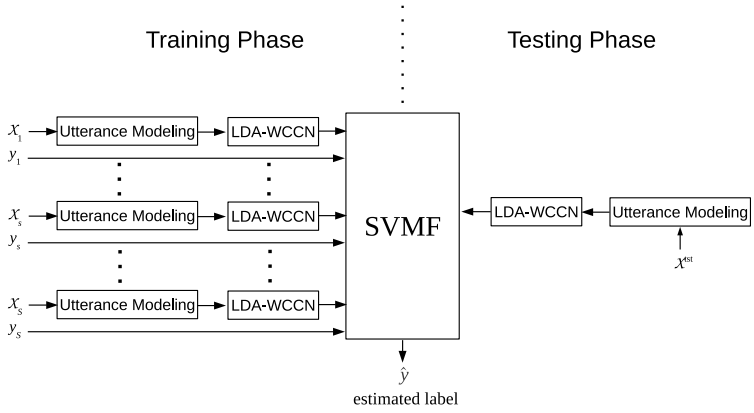

Figure 5.6: *The block-diagram of applied classification scheme NIST 2011 LRE and QCRI Arabic DRE experiments.*

technique, named feature Nuisance Attribute Projection (fNAP), is then applied on the features domain, similar to the approach proposed in [30].

Classification and calibration

The block-diagram of the applied classification scheme is shown in Figure 5.6. As can be interpreted from this figure, in the training phase, each utterance in the train dataset is converted to a vector using one of the utterance modeling approaches (ML, SMM, NMF, NFA, or i-vector) described in Sections 5.3.4, 5.3.3 and 5.4. Then, the obtained vectors representing the utterances are length normalized –such that their second norm equal to unity– and transformed using linear discriminant analysis (LDA), such that the ratio of the transformed between-class-scatter and the transformed within-class-scatter is maximized [31]. The number of discriminant dimensions in the applied LDA equals the number of categories minus one. The low-dimensional vectors are then transformed using within-class covariance normalization (WCCN) to transform the within-class covariance of the vector space to an identity matrix [32]. In doing so, directions of relatively high within-class variation will be attenuated, and thus prevented from dominating the space [32]. The projection matrices of LDA and WCCN are trained using the training data from all languages. Then, the obtained transformed vectors along with their corresponding language/dialect labels are used to train a scoring approach working based on simplified Von-Mises-Fisher distribution [27]. This scoring approach, labeled as SVMF in this chapter, is described in [27].

In the testing phase, the utterance modeling approach applied in the training phase is used to extract a vector from the utterance of an unseen speaker. Then the projection matrices of LDA and WCCN calculated in the training phase are applied to transform the obtained vector representing the test utterance to a low-dimensional space. Finally the trained SVMF uses the transformed vector to recognize the language/dialect of the test speaker. The SVMF score of the transformed test vector ν_{test} for the d^{th} language is obtained as follows

$$\mathcal{S}_d = \nu'_{test}\bar{\nu}_d, \tag{5.36}$$

where $\bar{\nu}_d$ denotes the mean of the transformed vectors for the d^{th} language in the training dataset.

To obtain well-calibrated scores on the evaluation dataset, linear logistic regression calibration [33, 34] is applied in the back-end. In this research, the FoCal Multiclass Toolkit [33] is applied to perform this calibration.

Performance Measure

In this experiment, the effectiveness of the proposed method is evaluated using log-likelihood-ratio cost (C_{llr}) [34, 35], which is also referred to as multi-class-cross-entropy in literature [36]. C_{llr} is an application-independent performance measure for recognizers with soft decision output in the form of log-likelihood-ratios. This performance measure, which has been adopted for use in the NIST speaker recognition evaluation, was initially developed for binary classification problems such as speaker recognition. It was extended to multi-class classification problems such as language recognition [34] as follows:

$$C_{\text{llr}} = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{|\kappa_d|}\sum_{k\in|\kappa_d|}-\log_2 P_{d,k} \tag{5.37}$$

$$P_{d,k} = \frac{\pi_d \mathcal{S}_{d,k}}{\sum_\jmath \pi_\jmath \mathcal{S}_{\jmath,k}}, \tag{5.38}$$

where is the κ_d is the subset of indices for test samples of class d, $|\kappa_d|$ is the total number of samples in the test set belonging to class d, π_d is the prior probability of d^{th} language class and subscript k denotes the indice for samples in the testing dataset.

C_{llr} ranges between zero and infinity. For a perfect classifier without errors C_{llr} equals to zero, otherwise it is a positive number. The reference level of C_{llr} for indicating the effectiveness of classifier is $\log_2 D$, e.g. for a two class recognition

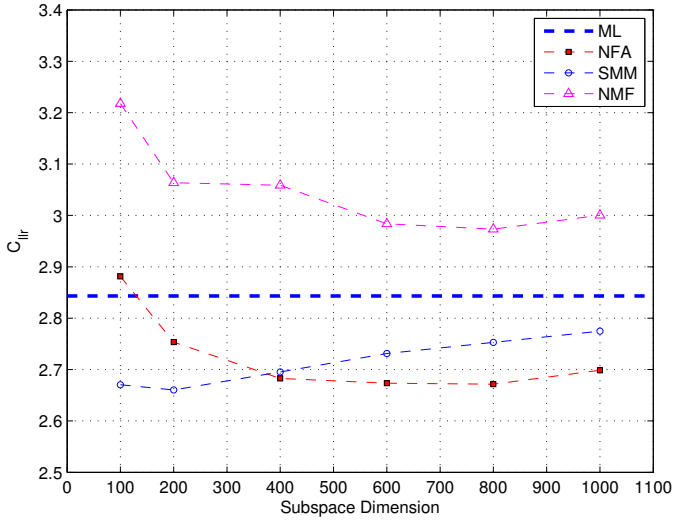

Figure 5.7: *The C_{llr} of language recognition using the proposed method and baseline systems versus subspace vector dimension.*

problem reference level is $\log_2 D = 1$. For a useful recognizer, $C_{\text{llr}} < \log_2 D$ and for poor input scores (poor $P_{d,k}$) $C_{\text{llr}} > \log_2 D$, indicating that it would be better to apply prior information rather than the recognizer [34].

In this research, we apply the FoCal Multiclass Toolkit [33] to calculate C_{llr}.

Comparison with Baseline Systems

Figure 5.7 shows the C_{llr} of language recognition for all utterances in testing dataset (regardless of utterance duration) using the proposed method and baseline systems versus the subspace vector dimension. This figure shows that the proposed method and the SMM increase the performance of language recognition compared to the ML weight supervector. It is also shown that the best results of the proposed method and the SMM are obtained at target dimension 800 and 200 respectively and the performance of the proposed method is robust against subspace dimension changes between dimensions 500 and 800.

For comparison purposes, all experiments on NIST 2011 LRE are performed using a computer with CPU model of Intel Xeon E5-1620 0 at 3.60GHz and 16 GB of RAM. Figure 5.8 shows the required computation time (elapsed

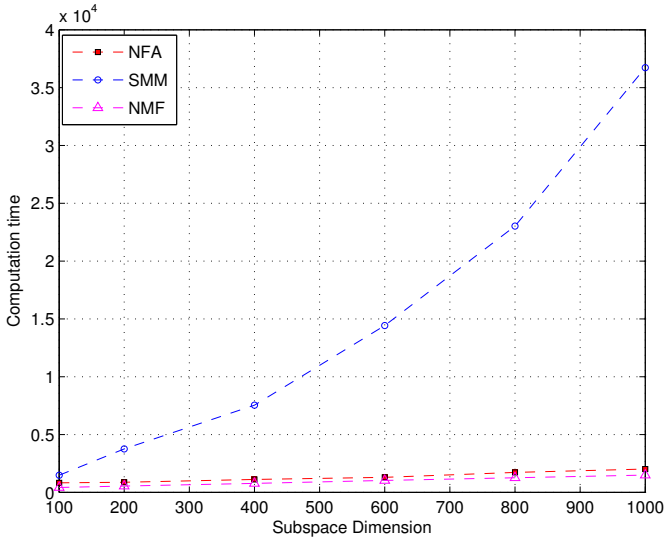

Figure 5.8: *The required computation time for estimating the subspace matrices using the proposed method and baseline systems versus subspace vector dimension.*

time) for estimating the subspace matrices using the proposed method and baseline systems versus subspace vector dimension. This figure shows that the required computation time for estimating the subspace matrices using the SMM is significantly higher than that of NFA and NMF especially for higher subspace dimensions. The required time for NFA and NMF grows linearly by increasing the subspace vector dimension, while this growth is cubic in the case of SMM.

Figure 5.9 shows the language recognition performance using the proposed method and baseline systems in different utterance length conditions. This bar chart demonstrates the results of NMF, SMM and NFA in their best subspace dimension. This figure shows that the proposed method and SMM improve the ML estimations at 3s, 10s, and 30s utterance length conditions. The obtained relative improvements [37] by the NFA compared to the ML baseline system in 3s, 10s and 30s conditions are 2.7%, 8.1%, and 11.6% respectively.

Fusion with i-vector Framework

The goal of this research is improving the recognition accuracy of the state-of-the-art i-vector system. The applied baseline i-vector system in this research is

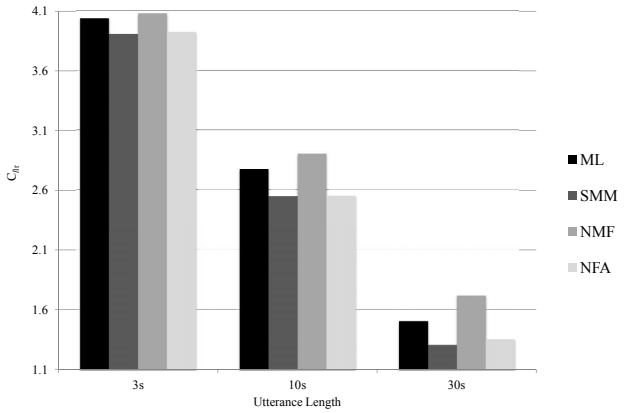

Figure 5.9: *The C_{llr} of language recognition using the proposed method and baseline systems in different utterance length conditions.*

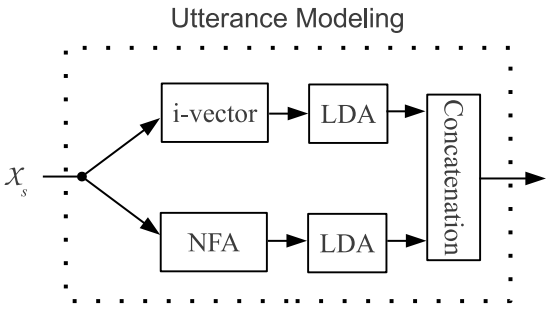

Figure 5.10: *The block-diagram of utterance modeling in intermediate-level fusion.*

the same as the ivec 1 subsystem of the MITLL NIST 2011 LRE submission [27]. The ivec 1 subsystem achieved the highest performance in comparison to other acoustic and phonotactic subsystems of the MITLL submission. To improve this system, an intermediate-level fusion of i-vectors and NFA subspace vectors is proposed. The block-diagram of the applied classification procedure in training and testing phases is the same as Figure 5.6. However, the utterance modeling blocks are replaced with the illustrated block in Figure 5.10. As shown in this figure, each i-vector, which is of dimension 600, is projected to a

Table 5.1: *The C_{llr} of language recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors.*

Method	3s	10s	30s
i-vector	3.39	1.71	0.775
i-vector-ML	3.32	1.70	0.773
i-vector-NMF	3.31	1.66	0.762
i-vector-SMM	3.30	1.62	0.725
i-vector-NFA	3.28	1.60	0.717

low-dimensional (the number of categories minus one) space using LDA. The LDA transformation matrix is calculated using all i-vectors in the training dataset. The same procedure is performed on the NFA subspace vectors. Then the obtained low-dimensional vectors are concatenated to form a new vector. Then, the obtained vectors modeling the utterances are applied to identify the utterance language using the classification procedure of Figure 5.6, where LDA and WCCN are applied for session variability compensation and SVMF is used as a classifier.

Table 5.1 lists the i-vector based system and obtained results after the proposed intermediate-level fusion. The intermediate-level fusions of the i-vector framework with NMF, SMM and NFA are performed using the best subspace dimension of these methods. As can be seen in this table, the obtained relative improvements [37] by this fusion compared to the state-of-the-art i-vector based recognizer in 3s, 10s, and 30s conditions are 3.33%, 6.23%, and 7.45% respectively.

5.6.2 QCRI Arabic DRE

Database

The Qatar computing research institute (QCRI) Arabic DRE corpus consists of Broadcast News, in four dialects; Egyptian, Levantine, Gulf, and Modern Standard Arabic (MSA). Data recordings were done using satellite cable sampled at 16kHz. The Aljazeera channel is the main source for the collected data. The recordings have been segmented into a wide range of durations to avoid speaker overlap, and avoid any non-speech parts such as music and background noise. Table 5.2 lists the number of utterances in each category for training, development and evaluation datasets.

Table 5.3 lists the number of utterances in different time durations.

Table 5.2: *The number of utterances for each dialect category in the QCRI corpus.*

Dialect	Training	Development	Evaluation
Egyptian	1116	463	139
Levantine	1074	186	132
Gulf	1181	221	218
MSA	1480	254	207
Total	**5051**	**1124**	**696**

Table 5.3: *The number of utterances in different durations in the QCRI corpus.*

Duration	Training	Development	Evaluation
shorter than 5s	723	141	97
5s-10s	754	156	103
10s-20s	968	225	123
20s-30s	649	153	100
30s-60s	835	207	102
Longer than 60s	366	115	41

UBM and Features

In the QCRI Arabic DRE experiment, the applied UBM has 512 mixtures and the feature extraction stage is based on a Shifted Delta cepstral representation. Speech is windowed at 20ms with a 10ms frame shift filtered through a Mel-scale filter bank. Each vector is then converted into a 56-dimensional vector following a SDC parameterization using a 7-1-3-7 configuration [29], and concatenated with the static cepstral coefficients. The SDC feature vectors are mean and variance normalized over each speech recording. The applied i-vectors in this experiment have 400 dimension.

Performance Measure

In this experiment, the effectiveness of the proposed method is evaluated using the percentage of incorrectly classified utterances (E_{ic}), which can be calculated using the following relation:

$$E_{\text{ic}} = \frac{\kappa_{\text{ic}}}{\kappa} \tag{5.39}$$

where κ_{ic} and κ denote the number of incorrectly classified utterances, and the total number of utterances in the test dataset respectively.

Table 5.4: *The E_{ic} of dialect recognition using the proposed method and baseline systems in QCRI Arabic DRE experiment (%).*

Method	Development	Evaluation
ML	31.9	33.5
NMF	31.2	32.6
SMM	36.9	34.0
NFA	30.1	30.7

Comparison

In this experiment, the same classification and calibration procedure of Section 5.6.1 is used, and the block-diagram of the applied classification scheme is shown in Figure 5.6. However, to calculate E_{ic}, rather that soft scores, we require hard decision, which is performed by maximizing over the obtained scores for each category.

Table 5.4 lists the E_{ic} of dialect recognition using the proposed method and baseline systems. In this experiment, SMM, NMF, and NFA have been tested over different target dimensions between 50 and 500, and Table 5.4 only includes the best results, which were obtained for target dimensions 400, 200, and 400 for NMF, SMM, and NFA respectively. As can be seen in this table, the NMF, and NFA subspace approaches improve the ML results in this experiment.

We also used the same intermediate-level fusion scheme described in Section 5.6.1 to improve the accuracy of the i-vector based system. Table 5.5 lists the E_{ic} of dialect recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors. As can be seen in this table, the average of E_{ic} over development and evaluation datasets for the i-vector framework and proposed fusion scheme are 19.65% and 15.5% respectively. Comparison of these values shows that the absolute and the relative improvements [37], obtained by intermediate-level fusion of the proposed method with the i-vector system are around 4%, and 21% respectively.

5.6.3 RATS LRE

Database

The Robust Automatic Transcription of Speech (RATS) P2 evaluation corpus is partially sourced from existing databases including

- Fisher Levantine conversational telephone speech (CTS).
- Callfriend Farsi CTS.

Table 5.5: *The E_{ic} of dialect recognition using the proposed method and baseline systems after intermediate-level fusion with i-vectors in QCRI Arabic DRE experiment (%).*

Method	Development	Evaluation
i-vector	19.6	19.7
i-vector-ML	15.9	15.8
i-vector-NMF	15.5	15.0
i-vector-SMM	16.4	15.9
i-vector-NFA	16.0	15.0

Table 5.6: *The number of utterances for each category in the RATS corpus.*

Language	Training	Development	Evaluation
Dar	3305	2733	184
Arle	46760	4023	1085
Urd	22775	4019	908
Pas	29605	4007	1032
Far	9006	3999	947
Non-Target	29208	9723	2518
Total	**140659**	**28504**	**6674**

• NIST LRE Data - Dari, Farsi, Pashto, Urdu and non-target languages.
New data, namely RATS Farsi, Urdu, Pashto, Levantine CTS, were also collected and added to the database. All recordings were retransmitted through eight different communication channels. The RATS goal is to categorize test set speech recordings into six different groups including five target languages, namely Dari (Dar), Arabic Levantine (Arle), Urdu (Urd), Pashto (Pas), Farsi (Far), and one non-target category which can be from 10 unknown languages. The RATS P2 evaluation corpus is divided into three disjoint databases namely training, development and evaluation. Table 5.6 lists the number of utterances in each category for training, development and evaluation datasets. The duration of all utterances in the training and development datasets is 120 seconds (s). Therefore, shorter duration speech signals have been created by cutting the original utterances after speech activity detection. The evaluation set speech signals has four different durations 120s, 30s, 10s and 3s.

Table 5.7: *The E_{ic} of dialect recognition using the proposed method and baseline systems in RATS LRE experiment (%).*

System Configuration	Evaluation Dataset			
	120s	30s	10s	3s
ML	14.0	32.1	49.3	61.9
NFA	11.0	25.2	42.1	58.7
i-vector	8.9	24.5	39.0	53.2
Fusion	8.1	22.5	35.5	46.6

UBM and Features

In this experiment, the applied UBM has 2048 mixtures, and the feature extraction stage used in this experiment is based on a Shifted Delta cepstral representation. Speech is windowed at 20ms with a 10ms frame shift filtered through a Mel-scale filter bank. Each vector is then converted into a 56-dimensional vector following a SDC parameterization using a 7-1-3-7 configuration [29], and concatenated with the static cepstral coefficients. Speech activity detection based on a Brno university of technology neural network implementation is then applied to remove the silence [38]. The applied i-vectors in this experiment have 600 dimension.

Classification

In this experiment, we applied a four-layer Deep belief nets (DBN) [39], where the first hidden layer consists of 1600 units, the second hidden layer consists of 200 units and the output layer has 6 units (the number of language categories).

Comparison

Table 5.7 lists the E_{ic} for the proposed method and baseline systems. The results of NMF and SMM are slightly worse than that of ML in this experiment, hence excluded from the table. The large number of utterances and highly degraded channels [40], which may rise the chance of falling into local minima, can be the reason of unsatisfactory results in SMM and NMF. As can be seen in this table, the average of E_{ic} over 120s, 30s, 10s, and 3s time conditions for the NFA and ML are 34.23% and 39.3% respectively. Therefore, the absolute improvement obtained by the proposed method compared to the baseline ML system is 5%. However, the accuracy of NFA, which works based on Gaussian weights, is lower than the i-vector based system, which works based on Gaussian means. This

concurs with previous studies demonstrating that GMM weight supervectors, which entail a lower dimension compared to Gaussian mean supervectors, carry less information than GMM means [14–16]. However, Gaussian weights provide a source of complementary information to the Gaussian means. Therefore, to enhance the accuracy of language recognition we apply a fusion of i-vectors and NFA vectors. The last row of Table 5.7 shows the fusion results obtained by concatenating i-vectors with NFA subspace vectors. As can be seen in this table, the average of E_{ic} over 120s, 30s, 10s, and 3s time conditions for the i-vector framework and proposed fusion scheme are 31.4% and 28.17% respectively. Comparison of these values shows that the absolute and the relative improvements [37] obtained by the proposed fusion are around 3% and 10% respectively. The improvement is more evident in the case of short utterances.

5.7 Conclusions

In this chapter, a new subspace method, non-negative factor analysis (NFA), for GMM weight adaptation has been introduced. The proposed approach applies a constrained factor analysis and suggests a new low-dimensional utterance representation. Evaluation on three different language/dialect recognition corpora, namely NIST 2011 LRE, RATS LRE and QCRI Arabic DRE, show that the proposed utterance representation scheme yields more accurate recognition results compared to ML re-estimation, SMM, and NMF approaches, while keeping the required computation time similar to NMF and considerably less than SMM. To improve the recognition accuracy of the state-of-the-art i-vector framework, an intermediate, or feature level fusion of i-vectors and proposed subspace vectors has been suggested. Experimental results show that the obtained relative improvements of the fusion scheme compared to i-vector frameworks are 6%, 20%, and 10% for NIST 2011 LRE, QCRI Arabic DRE, and RATS LRE.

5.8 Appendix I

The function to be maximized is

$$\Phi(\lambda, \mathbf{r}) = \bar{\gamma}'(\mathcal{X}) \log\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right) \tag{5.40}$$

The equality constraint is

$$\mathbf{1}\left(\mathbf{b} + \mathbf{L}\mathbf{r}\right) = 1 \tag{5.41}$$

By introducing a Lagrange multiplier we obtain

$$z(x) = \bar{\gamma}'(\mathcal{X}) \log(\mathbf{b} + \mathbf{L}\mathbf{r}) + \beta\,[1 - \mathbf{1}(\mathbf{b} + \mathbf{L}\mathbf{r})] \tag{5.42}$$

By differentiating Eq. 5.42 with respect to $\mathbf{r}$ and setting the result to 0 we obtain

$$\frac{[\bar{\gamma}(\mathcal{X})]'}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})]'}\mathbf{L} = \beta\mathbf{1}\mathbf{L} \tag{5.43}$$

Since $\mathbf{L}$ is a full rank matrix, we can drop it from both sides of Eq. 5.43.

$$\frac{[\bar{\gamma}(\mathcal{X})]'}{[\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})]'} = \beta\mathbf{1} \tag{5.44}$$

hence

$$\bar{\gamma}(\mathcal{X}) = \beta\,(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})) \tag{5.45}$$

Considering the equality constraint mentioned in Eq. 5.22 and multiplying with $\mathbf{1}$ on both sides of Eq. 5.45

$$\mathbf{1}\bar{\gamma}(\mathcal{X}) = \beta\mathbf{1}\,(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})) \tag{5.46}$$

or

$$\tau = \beta \tag{5.47}$$

Therefore,

$$\bar{\gamma}(\mathcal{X}) = \tau\,(\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})) \tag{5.48}$$

from which the Eq. 5.23 is obtained. Therefore, Eq. 5.23 is the analytical solution of the constrained optimization problem defined in Eq. 5.22.

Note that since τ and all elements of $\bar{\gamma}(\mathcal{X})$ in Eq. 5.48 are non-negative, the result of Eq. 5.23 keeps all elements of $\mathbf{b} + \mathbf{L}\mathbf{r}(\mathcal{X})$ non-negative as well.

5.9 References

[1] F. Biadsy, "Automatic Dialect and Accent Recognition and its Application to Speech Recognition," *Columbia University*, 2011.

[2] A. Hanani, "Human and computer recognition of regional accents and ethnic groups from British English speech," *University of Birmingham*, July 2012.

[3] Y. Muthusamy, E. Barnard, and R. Cole, "Reviewing automatic language identification," *Signal Processing Magazine, IEEE*, vol. 11, no. 4, pp. 33–41, 1994.

[4] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[5] R. G. Leonard and G. R. Doddington, "Automatic Language Identification.," *Technical Report RADC-TR-74-2007TI-347650, RADC/Texas Instruments, Inc., Dalas, TX*, 1974.

[6] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *The Journal of the Acoustical Society of America*, vol. 62, p. 708, 1977.

[7] H. A., R. M.J., and C. M.J., "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech and Language*, vol. 27, no. 1, pp. 59–74, 2013.

[8] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[9] W. M. Campbell, F. Richardson, and D. Reynolds, "Language recognition with word lattices and support vector machines," in *Proceedings of ICASSP*, 2007.

[10] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[12] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, pp. 857–860, 2011.

[13] M. Bahari, M. McLaren, H. Van hamme, and D. Van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Interspeech*, pp. 506–509, 2012.

[14] M. H. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," in *Proceedings ICASSP'2013*, pp. 7344–7348, 2013.

[15] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[16] X. Zhang, K. Demuynck, and H. Van hamme, "Rapid Speaker Adaptation in Latent Speaker Space with Non-negative Matrix Factorization," *Speech Communication*, 2013.

[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[18] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, and J. Cernocky, "Prosodic speaker verification using subspace multinomial models with intersession compensation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[19] O. Glembek, P. Matejka, L. Burget, and T. Mikolov, "Advances in phonotactic language recognition," *Interspeech'08*, pp. 743–746, 2008.

[20] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "iVector approach to phonotactic language recognition," in *Proc. of Interspeech*, pp. 2913–2916, 2011.

[21] M. Soufifar, S. Cumani, L. Burget, and J. Cernocky, "Discriminative classifiers for phonotactic language recognition with iVectors," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4853–4856, IEEE, 2012.

[22] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, no. 5, pp. 980–988, 2008.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal statistical Society-Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[24] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[25] J. A. Snyman, *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, vol. 97. Springer Science+ Business Media, 2005.

[26] M. M. Soufifar, L. Burget, O. Plchot, S. Cumani, and J. Cernocky, "Regularized Subspace n-Gram Model for Phonotactic iVector Extraction," in *Interspeech*, pp. 74–78, 2013.

[27] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," *Speaker Odyssey 2012*, pp. 209–215, 2012.

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

[29] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features.," in *INTERSPEECH*, 2002.

[30] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, pp. 1–6, IEEE, 2006.

[31] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification and Scene Analysis 2nd ed.," 1995.

[32] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, vol. 4, 2006.

[33] N. Brummer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores," *Tutorial and User Manual. Spescom DataVoice*, 2007.

[34] N. Brummer and D. van Leeuwen, "On calibration of language recognition scores," in *Proc. IEEE Odyssey Speaker and Language Recognition Workshop*, pp. 1–8, IEEE, 2006.

[35] N. Brummer, "Application-independent evaluation of speaker detection," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[36] L. J. Rodríguez-Fuentes, N. Brümmer, M. Penagarikano, A. Varona, M. Diez, and G. Bordel, "The Albayzin 2012 Language Recognition Evaluation Plan (Albayzin 2012 LRE)," 2012.

[37] E. D. Bolker and M. Mast, *Common Sense Mathematics*. Citeseerx, 2005.

[38] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a Speech Activity Detection System for the DARPA RATS Program.," in *INTERSPEECH*, 2012.

[39] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[40] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Proc. Odyssey*, 2012.

Chapter 6

Speaker age estimation using a fusion of the i-vector and NFA framesworks

This chapter is based on the following article:
1) Bahari, M.H., Van hamme (2014), "Speaker age estimation using a fusion of the i-vector and non-negative factor analysis frameworks," Pattern Recognition Letters, Elsevier (submitted).

6.1 Abstract

In this chapter, a new approach for age estimation from speech signals based on a hybrid architecture including the i-vector and non-negative factor analysis (NFA) frameworks is proposed. In this method, each utterance is modeled by its corresponding i-vector and NFA vector. Then, two subsystems are proposed such that the first subsystem is based on i-vectors and the second subsystem works based on feature-level fusion of NFA vectors and i-vectors. In both subsystems, least squares support vector regression (LSSVR) is applied for regression. Finally, score-level fusion of the developed subsystems is considered. The proposed method is trained and tested on telephone conversations of the National Institute for Standard and Technology (NIST) 2010 and 2008 speaker recognition evaluation databases. Evaluation results show that the proposed method yields lower mean absolute estimation error and higher Pearson correlation coefficient between chronological speaker age and estimated speaker age compared to different conventional schemes.

6.2 Introduction

Automatic identification of a person's characteristics, such as gender, age, language/dialect, and psychological state, from speech signals has a wide range of commercial applications such as interactive voice response systems, targeted advertising, service customization, medical care, multimedia retrieval, forensic softwares and natural human-machine interaction [1–3]. This technology can also guide ambient assisted living and smart home systems to automatically adapt to different user needs [3]. In this research, we focus on speaker age estimation, which is an important ingredient of speaker profiling systems and behavioral informatics.

Experimental studies reveal major effects of vocal aging on the speech signal such as lowered speaking rate and increased jitter and shimmer [4], and has shown to negatively influence speaker recognition performance [5]. Such age dependent factors can be used as acoustic cues in automatic speaker age estimation. However, the relation of these acoustic cues with speaker age is usually complex and affected by many other factors such as speech content, language, gender, weight, height, emotional condition, smoking and drinking habits [4, 6, 7]. Furthermore, in many practical cases we have no control over the available speech duration, content, language, environment, recording device and channel conditions, etc..

Studies on the influence of ageing on voice started at the late 1950s [8]. However, the first automatic speaker age recognition systems were developed around four

decades later in the early 2000s [9–12]. During this decade, many different techniques, mostly inspired from automatic speaker and language recognition fields, have been suggested for categorizing speakers based on their age groups. For example, using Gaussian Mixture Model (GMM) mean supervectors and Support Vector Machine (SVM) [13–15], nuisance attribute projection [16], anchor models [16] and parallel phoneme recognizers [17]. The age sub-challenge of Interspeech 2010 paralinguistic challenge provided a forum for presenting state of the art methods in speaker age group classification [18]. Participants of the age sub-challenge tried to categorize speakers of the "aGender" corpus into four age groups—7 to 14 (Child), 15 to 24 (Youth), 25 to 54 (Adult) and 55 to 80 (Senior) years old— using their telephone speech signals. In this sub-challenge GMM mean supervector [19], GMM weight supervector [20], Maximum-Mutual-Information (MMI) training [21] and fuzzy SVM modeling [22] have been suggested to enhance acoustic modeling quality. A brief overview of different proposed methods in this sub-challenge is presented in [3], which also introduces an age group recognition approach using acoustic and prosodic level information fusion.

One effective approach to age estimation from speech involves modeling speech recordings with Gaussian Mixture Model (GMM) mean supervectors to use them as features in Support Vector Regression (SVR) [1, 23]. Similar Support Vector Machine (SVM) techniques have been successfully applied to different speech processing tasks such as speaker recognition [24]. While effective, GMM mean supervectors are of a high dimensionality resulting in high computational cost and difficulty in obtaining a robust model in the context of limited data. Consequently, dimension reduction through PCA-based methods has been found to improve performance in age estimation from GMM mean supervectors [1]. In the field of speaker and language recognition, recent advances using the i-vector framework [25, 26], which provide a compact representation of an utterance in the form of a low-dimensional feature vector, have increased the classification accuracy considerably. i-vectors successfully replaced GMM mean supervectors in speaker age estimation as well [27].

We have recently introduced a new framework for adaptation and decomposition of GMM weights based on a factor analysis similar to that of the i-vector framework [28]. In this method, namely non-negative factor analysis (NFA), the applied factor analysis is constrained such that the adapted GMM weights are non-negative and sum to unity. This method, which yields a new low-dimensional utterance representation approach, was applied to speaker and language/dialect recognition successfully [28–30].

In this chapter, we propose a new approach for speaker age estimation based on a hybrid architecture of NFA and i-vector frameworks exploiting the available information in Gaussian means and Gaussian weights. This architecture consists

of two subsystems based on i-vectors and NFA vectors. The first subsystem uses i-vectors as features and applies least squares support vector regression (LSSVR) for regression. In the second subsystem, feature level fusion of i-vectors and NFA vectors is applied and LSSVR is used for regression. Finally, the estimated ages using each subsystem are fused to enhance the age estimation accuracy of each subsystem. Evaluation on the NIST 2010 and 2008 SRE databases shows that the proposed method improves both mean absolute error and correlation coefficient considerably.

The rest of this chapter is organized as follows. In Section 6.3 the problem of speaker age estimation and different conventional approaches addressing this issue are described. In section 6.4, the proposed approach is elaborated. Section 6.5 explains our experimental setup. The evaluation results are presented and discussed in section 6.6. The chapter ends with conclusions in section 6.7.

6.3 Age Estimation from Speech

In speaker age estimation, we are given a training dataset of speech recordings $S^{\text{tr}} = \{(\mathcal{X}_1, y_1), \ldots, (\mathcal{X}_s, y_s), \ldots, (\mathcal{X}_S, y_S)\}$, where $\mathcal{X}_s$ denotes the s^{th} utterance of the training dataset, and y_s denotes the s^{th} utterance of the training dataset and its corresponding speaker age, respectively. The goal is to design an estimator function $g(\mathcal{X})$, such that for an utterance of an unseen speaker $\mathcal{X}^{\text{tst}}$, the actual speaker age is predicted accurately.

6.3.1 Baseline Approaches

In this chapter, we use three baseline approaches with which we compare our proposed regression techniques:
Prior: The most basic choice for the estimator function is the average age of the training data, $g(x^{\text{tst}}) = \frac{1}{S}\sum_s y_s$. This estimator, labeled as *prior* in the rest of this chapter, intuitively provides a reference level of accuracy.
GMM-R: Different methods have been introduced to reach an effective speaker age estimation [1, 6, 23]. For example, Bocklet *et al.* introduced GMM-R to estimate the age of children from GMM mean supervectors derived from their utterances [23]. Given an utterance, Maximum A Posteriori adaptation (MAP) is applied to adapt a Universal Background Model (UBM) to the speech characteristics of the speaker [24]. The component means of the obtained GMM are then extracted and concatenated to form a GMM mean

supervector representing the utterance. Finally, an SVR is applied as a function approximator to estimate the speakers' age.

GMM-PCA-R and **GMM-WPPCA-R:** The approach of GMM-R was adopted and extended by Dobry *et al.* [1] by applying dimension reduction techniques to the supervector. Methods such as Principal Component Analysis (PCA) and Weighted-Pairwise PCA (WPPCA) were applied and investigated. It was concluded that WPPCA, which is a supervised dimensionality reduction approach working based on nuisance attribute projection [1], yields more accurate results. These speaker age estimators, labeled GMM-PCA-R and GMM-WPPCA-R, are used as contrastive baseline systems in this chapter.

i-vector-R: The i-vector framework is a subspace approach for GMM mean adaptation based factor analysis. In [27] GMM mean supervectors were replaced by i-vectors, which are a compact representation of an utterance in the form of a low-dimensional feature vector. This method, labeled as i-vector-R in this chapter, is used as a baseline system and as the first subsystem of the proposed approach.

6.4 System Description

In this section, the main components of the proposed method, namely LSSVR, the i-vector and NFA frameworks, are described. Then, the proposed method in training and testing phases is elaborated.

6.4.1 Regression using LSSVR

Least Squares Support Vector Machine (LSSVM), which is a variant of SVM, was introduced by Suykens and Vandewalle [31]. It is employed as a machine learning tool for classification, clustering and regression tasks. Compared to SVM, LSSVM benefits from a faster training process because the quadratic programming problem of SVM is reduced to that of solving a system of linear equations. Furthermore, the LSSVM formulation involves fewer tuning parameters [32]. A continuous function can be fitted to the training data with a Least Squares Support Vector Regressor (LSSVR), a technique which shares many of the advantages of LSSVM classification.

In a typical regression problem a training dataset $S^{\text{tr}} = \{(\mathbf{w}_1, y_1), \ldots, (\mathbf{w}_s, y_s), \ldots, (\mathbf{w}_S, y_S)\}$, where $\mathbf{w}_s$ denotes a vector of observed features of the data item and y_s denotes the model input and corresponding

output of the s^{th} data point respectively. The objective of the regression analysis is to determine a function $f(\mathbf{w})$, so as to predict the desired outputs accurately. In the primal form of LSSVR, which is the same as SVR, the following relation is considered for $f(\mathbf{w})$

$$f(\mathbf{w}) = \varpi' \Phi(\mathbf{w}) + z. \tag{6.1}$$

In LSSVR, a least squares loss function is applied instead of Vapnik's ϵ-insensitive loss function to simplify the formulations to minimize

$$\frac{1}{2} \|\varpi\|^2 + \frac{1}{2} \vartheta \sum_{s=1}^{S} e_s^2 \tag{6.2}$$

subject to

$$y_s = \varpi' \Phi(\mathbf{w}_s) + z + e_s, \tag{6.3}$$

where ϑ is an error cost factor and $e_s \in \mathbb{R}$ are error variables.

For high dimensional data this optimization problem can be solved more efficiently by introducing the Lagrangian variables ν and minimizing the following dual cost function [31]

$$\begin{aligned} \Psi(\varpi, z, e, \nu) =& \frac{1}{2} \|\varpi\|^2 + \frac{1}{2} \vartheta \sum_{s=1}^{S} e_s^2 \\ & - \sum_{s=1}^{S} \nu_s \{\varpi' \Phi(\mathbf{w}_s) + z + e_s - y_s\}. \end{aligned} \tag{6.4}$$

One can solve this optimization problem directly by taking the partial derivative of Ψ with respect to ϖ, z, e and ν and setting the results to zero which leads to solving a linear system of equations. Inserting the obtained results into 6.1 leads to the regression function

$$\begin{aligned} f(\mathbf{w}) &= \sum_{s=1}^{S} \nu_s \langle \Phi(\mathbf{w}_s), \Phi(\mathbf{w}) \rangle + z \\ &= \sum_{s=1}^{S} \nu_s K(\mathbf{w}_s, \mathbf{w}) + z, \end{aligned} \tag{6.5}$$

where $K(\mathbf{w}_s, \mathbf{w})$ is the kernel function and ν and z are the solution to optimization problem 6.4.

LSSVR has two advantages and one drawback compared to SVR. The first advantage of LSSVR is that its model training is faster as its dual form corresponds to solving a linear system which involves less computation time compared to a quadratic programming problem of SVR. The second advantage is that the LSSVR is faster to tune as its formulation involves fewer hyperparameters to tune (the minimal error margin ϵ is not used here). A drawback of this simplification is the loss of sparseness, which has been highlighted in literature [33, 34].

In this research, the LSSVR model training and testing is implemented using LSSVMlab [31] and the hyperparameters of the LSSVR are tuned on the training set using the N-fold cross validation technique.

6.4.2 Utterance Modeling

The first step toward model-based speaker age estimation is converting variable-duration speech signals into fixed-dimensional vectors, which is performed by fitting a GMM to acoustic features extracted from each speech signal. The parameters of the obtained GMMs characterize the corresponding utterance.

Due to a lack of data, fitting a separate GMM for a short utterance can not be performed accurately, especially in the case of GMMs with a high number of Gaussians. Therefore, parametric utterance adaptation methods are usually applied to adapt a universal background model (UBM) to characteristics of utterances in training and testing databases. In this chapter, the i-vector framework for adapting UBM means and the NFA framework for adapting UBM weights are applied.

Universal Background Model and Adaptation

Consider a UBM with the following likelihood function for $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_\tau\}$.

$$p(\mathbf{x}_t|\lambda) = \sum_{c=1}^{C} b_c p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$$

$$\lambda = \{b_c, \mu_c, \mathbf{\Sigma}_c\},\ c = 1, \ldots C, \tag{6.6}$$

where $\mathbf{x}_t$ is the acoustic vector at time t, b_c is the mixture weight for the c^{th} mixture component, $p(\mathbf{x}_t|\mu_c, \mathbf{\Sigma}_c)$ is a Gaussian probability density function

with mean μ_c and covariance matrix $\mathbf{\Sigma}_c$, and C is the total number of Gaussians in the mixture. The parameters of the UBM $-\lambda-$ are estimated on a large amount of training data from speakers of different ages.

i-vector Framework

One effective method for speaker age estimation involves adapting UBM means to the speech characteristics of the utterance. The adapted GMM means are subsequently extracted and concatenated to form Gaussian mean supervectors [1, 23]. This method has been shown to provide a good level of performance [1, 23]. Recent progress in this field, however, has found an alternate method of modeling GMM mean supervectors that provides superior recognition performance [7]. This technique referred to as total variability modeling [25] assumes the GMM mean supervector, $\mathbf{m}$, can be decomposed as

$$\mathbf{m} = \mathbf{u} + \mathbf{Tv}, \tag{6.7}$$

where $\mathbf{u}$ is the mean supervector of the UBM, $\mathbf{T}$ spans a low-dimensional subspace (400 dimensions in this work) and $\mathbf{v}$ are the factors that best describe the utterance-dependent mean offset $\mathbf{Tv}$. The vector $\mathbf{v}$ is treated as a latent variable with a standard normal prior and the so-called i-vector is its maximum-a-posteriori (MAP) point estimate. The subspace matrix $\mathbf{T}$ is estimated via maximum likelihood in a large training dataset. An efficient procedure for training $\mathbf{T}$ and for MAP adaptation of i-vectors can be found in [35]. In the total variability modeling approach, i-vectors are the low-dimensional representation of an audio recording that can be used for classification and estimation purposes.

The NFA Framework

NFA is a new framework for adaptation and decomposition of GMM weights based on a constrained factor analysis [28]. This new low-dimensional utterance representation approach was applied to speaker and language/dialect recognition tasks successfully [28–30].

The basic assumption of this method is that for a given utterance, the adapted GMM weight supervector can be decomposed as follows

$$\mathbf{w} = \mathbf{b} + \mathbf{Lr}, \tag{6.8}$$

where $\mathbf{b}$ is the UBM weight supervector (2048 dimensional vector in this work). $\mathbf{L}$ is a matrix of dimension $C \times \varkappa$ spanning a low-dimensional subspace. $\mathbf{r}$ is

a low-dimensional vector that best describes the utterance-dependent weight offset $\mathbf{Lr}$.

In this framework, neither subspace matrix $\mathbf{L}$ nor subspace vector $\mathbf{r}$ are constrained to be non-negative. However, unlike the i-vector framework, the applied factor analysis for estimating the subspace matrix $\mathbf{L}$ and the subspace vector $\mathbf{r}$ is constrained such that the adapted GMM weights are non-negative and sum up to one. The procedure of calculating $\mathbf{L}$ and $\mathbf{r}$ involves a two-stage algorithm similar to EM. In the first stage, $\mathbf{L}$ is assumed to be known, and we update $\mathbf{r}$. Similarly, in the second stage, $\mathbf{r}$ is assumed to be known and we update $\mathbf{L}$.

The subspace matrix $\mathbf{L}$ is estimated over a large training dataset. It is then used to extract a subspace vector $\mathbf{r}$ for each utterance in train and test datasets. The obtained subspace vectors representing the utterances in train and test datasets are used to estimate the age of speakers in this chapter.

6.4.3 System Architecture

The principle of the proposed age estimation approach in training phase is illustrated in Figure 6.1. As it can be interpreted from this figure, in the first estimator, each utterance of the training dataset is converted to an i-vector. Then, the obtained vectors along with their corresponding chronological speaker age are used to train the LSSVR 1. In the second estimator, each utterance is converted to a NFA vector and an i-vector. Then the obtained i-vector is concatenated with the NFA vector (after normalization) to form a longer vector (feature-level fusion). Finally, the obtained vectors along with their corresponding chronological speaker age are used to train the LSSVR 2.

Figure 6.2 shows the block-diagram of the proposed method in development and testing phases. In the development phase, LSSVR 3 is trained to fuse the results of estimator 1 and 2. To perform this fusion, first the age of each utterance of the development dataset is estimated using the trained estimators 1 and trained estimator 2 simultaneously. In estimator 1, the utterances are converted to i-vectors and then fed into the trained LSSVR 1 to estimate the age of speaker. In estimator 2, the utterances are converted to i-vectors and NFA vectors and then these two vectors are normalized and concatenated to form a longer vector, which is fed into the trained LSSVR 2 to estimate the age of speaker. Finally, the estimated age of each utterance using estimators 1 and 2 are concatenated to form a two dimensional vector. The obtained vectors along with their corresponding chronological speaker age are used to train LSSVR 3.

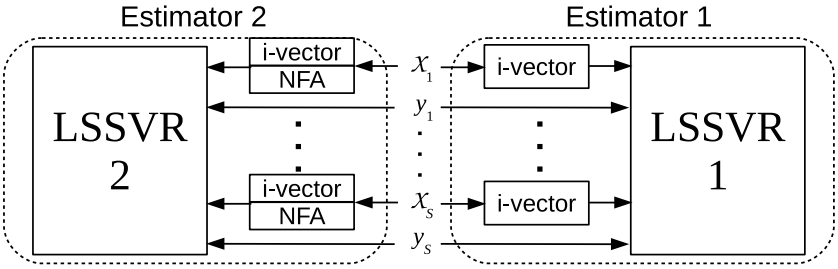

Figure 6.1: *The block-diagram of the proposed speaker age estimation approach in training phase.*

In testing phase, first estimator 1 and 2 are used to estimate the age of the utterance of an unseen speaker. Then, the output of estimator 1 and estimator 2 are concatenated and used in the trained LSSVR 3 to estimate the age of test speaker.

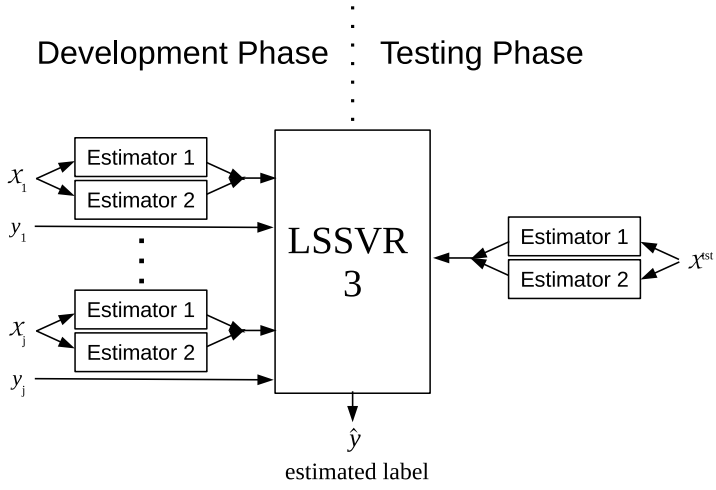

Figure 6.2: *The block-diagram of the proposed speaker age estimation approach in development and testing phases.*

6.5 Experimental Setup

6.5.1 Database

The National Institute of Standards and Technology (NIST) have held annual or biannual Speaker Recognition Evaluations (SRE) for the past two decades. With each SRE, a large database of telephone conversations (and more recently microphone speech) are released along with an evaluation protocol. These conversations typically last five minutes and originate from a large number of participants for whom meta data is recorded—including participant age and language. The NIST databases where chosen for this work due to the large number of speakers meeting the i-vector and NFA frameworks requirement for a considerable amount of data to estimate subspace matrices accurately. In our experiments, the parameters of UBM is estimated on a database including over $30,000$ speech recordings sourced from NIST 2004–2006 SRE corpora. The procedure of obtaining the applied UBM is presented in [36].

For the purpose of age estimation, telephone recordings from the common protocols of the NIST 2010 and 2008 SRE databases are used for training, development and testing. The core protocol, short2-short3, from the 2008 database contains 3999 telephone recordings for 1336 speakers for whom the age is known. Similarly, the extended core-core protocol of the 2010 database contains 5634 telephone speech segments from 445 speakers. There is no overlap between speech recordings extracted from the NIST 2010 and NIST 2008 SRE databases. Therefore, NIST 2008 SRE is used for testing. Among all utterances of NIST 2010 SRE, 150 utterances were used for development and the rest were used for training.

Figure 6.3 illustrates the age histograms of male and female speakers in the NIST 2010 and 2008 SRE databases.

6.5.2 Performance Metric

The effectiveness of the applied methods is evaluated using the Mean Absolute Error (E_{ma}) of the estimated speakers' age and Pearson's correlation coefficient (ρ) between the chronological speakers' age and the estimated speakers' age. The measure E_{ma} is calculated using:

$$E_{\text{ma}} = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \left| \hat{y}_k - y_k \right|, \tag{6.9}$$

where $\hat{y}_k$ and y_k are the estimated and the chronological age of the k^{th} utterance of the testing dataset respectively. κ is the total number of utterances in the

testing dataset. Further,

$$\rho = \frac{1}{\kappa - 1} \sum_{k=1}^{\kappa} \left(\frac{\hat{y}_k - \mu_{\hat{y}}}{\sigma_{\hat{y}}} \right) \left(\frac{y_k - \mu_y}{\sigma_y} \right), \tag{6.10}$$

where $\mu_{\hat{y}}$ and $\sigma_{\hat{y}}$ are the mean and the standard deviation of the speakers' estimated age respectively. Similarly μ_y and σ_y denote the mean and the standard deviation of the speakers' chronological age respectively.

6.6 Results and Discussion

This section presents the evaluation results of the baseline systems and compares them to the proposed age estimation system.

Acoustic features consist of 20 Mel-Frequency Cepstrum Coefficients (MFCCs) including appended energy with their first and second order derivatives, forming a 60 dimensional acoustic feature vector. In both cases, a hamming window is used and the sampling rate, frame rate, frame size and number of Mel frequency channels are 8000 Hz, 100 Hz, 0.02 s and 30 respectively. To have more reliable features, Wiener filtering, speech activity detection [37] and feature warping [38] have been considered.

6.6.1 Baseline Systems Results

In this section, the performance of the baseline systems, namely prior, GMM-R, GMM-PCA-R, GMM-WPPCA-R and i-vector-R are investigated.

In this experiment, PCA and WPPCA have been tested over different target dimensions between 100 and 1000. Table 6.1 only includes the best results, which were obtained for target dimensions 200 and 300 for GMM-PCA-R and GMM-WPPCA-R respectively. The dimensionality of the i-vectors is 400, the same as in [7].

Results in table 6.1 indicate that the GMM-R system is remarkably more accurate than the prior system. This shows that the GMM supervectors contain speaker information including age. Table 6.1 also shows that the PCA and WPPCA based systems outperform the GMM-R system, thus demonstrating the benefit of dimension reduction of the GMM supervectors prior to regression. Unlike [1] our experiments do not show a remarkable advantage for using WPPCA over PCA.

Table 6.1: *The E_{ma} (in years) and ρ of male and female speakers' age estimation for the baseline systems.*

System Configuration	Memale		Female	
	E_{ma}	ρ	E_{ma}	ρ
Prior	9.34	0	10.39	0
GMM-R	8.24	0.48	8.02	0.60
GMM-PCA-R	7.91	0.48	7.7	0.60
GMM-WPPCA-R	7.95	0.48	7.79	0.59

6.6.2 NFA Framework

To find the best subspace vector dimension in the NFA framework, we focus on estimator 2, however i-vector modeling of this estimator is completely ignored, i.e. utterances are converted to NFA vectors and after normalization are fed into LSSVR. Table 6.2 lists the E_{ma} of the estimated age and the ρ between the chronological speakers' age and the estimated speakers' age for different target dimensions respectively. As it can be seen in these figures, the best results are achieved at dimension 300. Therefore, in the rest of experiments, we use this NFA subspace dimension. The results of Tables 6.2 and 6.1 show that the NFA framework yields lower estimation accuracy compared to conventional approaches working based on Gaussian means. Previous studies [3, 28–30, 39, 40] show that GMM weights, which entail a lower dimension compared to Gaussian mean supervectors, carry less, yet complementary, information to GMM means. For example, Zhang *et al.* applied GMM weight adaptation in conjunction with mean adaptation for a large vocabulary speech recognition system to improve the word error rate [39]. Li *et al.* investigated the application of GMM weight supervectors in speaker age group recognition and showed that score-level fusion of classifiers based on GMM weights and GMM means improves recognition performance [3]. In [40] the feature level fusion of i-vectors, GMM mean supervectors, and GMM weight supervectors is applied to improve the accuracy of accent recognition. Therefore, in this study we considered feature-level and score-level fusion of i-vector and NFA frameworks to exploit the available information in both Gaussian means and Gaussian weights.

6.6.3 Proposed Method

The results of the proposed method for speakers' age estimation are presented in this section.

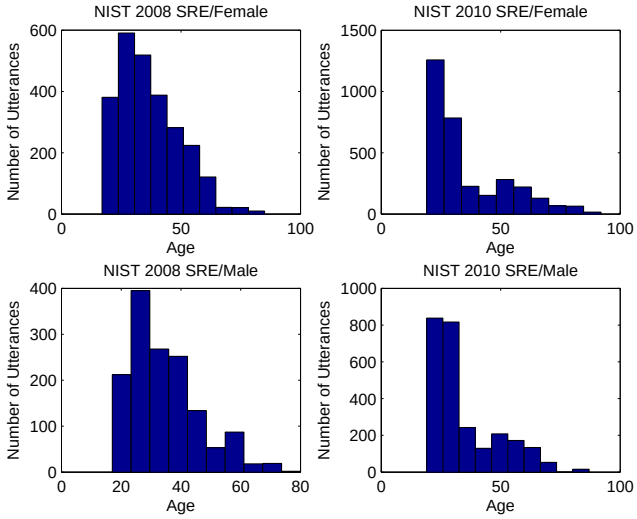

Figure 6.3: *Age histogram of telephone speech utterances for NIST 2010 and 2008 SRE Databases.*

The ρ and E_{ma} of age estimation using the proposed approach (after score-level fusion) are 0.593 and 7.38 respectively. Therefore, the proposed method improves ρ by 6%, 6% and 7% relative to GMM-R, GMM-PCA-R, and GMM-PCA-R respectively. The E_{ma} is also improved by 26%, 9%, 5% and 6% relative to Prior, GMM-R, GMM-PCA-R and GMM-PCA-R respectively.

To study the effect of the applied score-level and feature-level fusions, we also calculate the performance of estimators 1 and 2 before the fusion separately. Table 6.3 lists the E_{ma} of the estimated age and the ρ between the chronological speakers' age and the estimated speakers' age using the estimators 1 and 2 before

Table 6.2: *The E_{ma} (in years) and ρ of speakers' age estimation for NFA framework in different subspace dimensions.*

Subspace Dimension	E_{ma}	ρ
100	8.29	0.48
200	8.28	0.50
300	8.20	0.51
400	8.28	0.51
500	8.27	0.50

Table 6.3: *The E_{ma} (in years) and ρ of speakers' age estimation for estimators 1 and 2.*

System Configuration	E_{ma}	ρ
Estimator 1	7.63	0.58
Estimator 2	7.55	0.59
Proposed Method	7.38	0.593

fusion respectively. This table shows that estimator 2 is slightly more accurate than estimator 1 and thus demonstrating the benefit of feature-level fusion of i-vectors and NFA vectors. It also shows that the relative improvements of E_{ma} and ρ obtained by the proposed method compared to estimator 1 (state-of-the-art i-vector framework) are about 3.3% and 2.2% respectively.

6.7 Conclusions

In this chapter, utterances were modeled using the i-vector and NFA frameworks and a hybrid architecture of these approaches and LSSVR was developed to address the speaker age estimation problem. For the evaluation, telephone utterances of NIST 2010 and 2008 SRE databases have been used. Experimental results show that the accuracy of the proposed approach improves both mean absolute estimation error of estimation and Pearson correlation coefficient between chronological speaker age and estimated speaker age compared to different conventional schemes and the state-of-the-art i-vector framework.

6.8 Acknowledgements

This work is supported by the European Commission as a Marie-Curie ITN-project (FP7-PEOPLE-ITN-2008), namely Bayesian Biometrics for Forensics (BBfor2), under Grant Agreement number 238803.

We would like to thank Mitchell McLaren and David van Leeuwen for providing the i-vectors of this study.

6.9 References

[1] G. Dobry, R. M. Hecht, M. Avigal, and Y. Zigel, "Supervector dimension reduction for efficient speaker age estimation based on the acoustic speech

signal," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 7, pp. 1975–1985, 2011.

[2] D. C. Tanner and M. E. Tanner, *Forensic aspects of speech patterns: voice prints, speaker profiling, lie and intoxication detection.* Lawyers and Judges Publishing, 2004.

[3] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[4] S. Schotz, *Perception, analysis and synthesis of speaker age*, vol. 47. Citeseer, 2006.

[5] F. Kelly, A. Drygajlo, and N. Harte, "Speaker verification in score-ageing-quality classification space," *Computer Speech & Language*, 2013.

[6] M. H. Bahari and H. Van hamme, "Speaker age estimation and gender detection based on supervised non-negative matrix factorization," in *Proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pp. 1–6, 2011.

[7] M. H. Bahari and H. Van hamme, "Speaker age estimation using hidden markov model weight supervectors," in *11th IEEE Int. Conf. Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, 2012.

[8] E. D. Mysak, "Pitch and duration characteristics of older males," *Journal of Speech, Language and Hearing Research*, vol. 2, no. 1, p. 46, 1959.

[9] S. E. Linville, *Vocal aging.* Singular Thomson Learning, 2001.

[10] C. Muller, F. Wittig, and J. Baus, "Exploiting speech for recognizing elderly users to respond to their special needs," in *Proc. 8th European Conf. Speech Communication and Technology (Eurospeech)*, pp. 1305–1308, 2003.

[11] N. Minematsu, M. Sekiguchi, and K. Hirose, "Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. I–137, 2002.

[12] I. Shafran, M. Riley, and M. Mohri, "Voice signatures," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 31–36, 2003.

[13] D. Mahmoodi, A. Soleimani, H. Marvi, F. Razzazi, M. Taghizadeh, and M. Mahmoodi, "Age estimation based on speech features and support vector machine," in *3rd Computer Science and Electronic Engineering Conference*, pp. 60–64, 2011.

[14] C.-C. Chen, P.-T. Lu, M.-L. Hsia, J.-Y. Ke, and O.-C. Chen, "Gender-to-age hierarchical recognition for speech," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, pp. 1–4, 2011.

[15] C. van Heerden, E. Barnard, M. Davel, C. van der Walt, E. van Dyk, M. Feld, and C. Muller, "Combining regression and classification methods for improving automatic speaker age recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 5174–5177, 2010.

[16] G. Dobry, R. Hecht, M. Avigal, and Y. Zigel, "Dimension reduction approaches for svm based speaker age estimation," in *Proc. Interspeech*, pp. 2031–2034, 2009.

[17] F. Metze, J. Ajmera, R. Englert, U. Bub, F. Burkhardt, J. Stegmann, C. Muller, R. Huber, B. Andrassy, J. Bauer, and B. Littel, "Comparison of four approaches to age and gender recognition for telephone applications," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. IV–1089, 2007.

[18] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan, "The interspeech 2010 paralinguistic challenge," in *Proc. Interspeech*, pp. 2794–2797, 2010.

[19] T. Bocklet, G. Stemmer, V. Zeissler, and E. Noth, "Age and gender recognition based on multiple systems early vs. late fusion," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2830–2833, 2010.

[20] R. Porat, D. Lange, and Y. Zigel, "Age recognition based on speech signals using weights supervector," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2814–2817, 2010.

[21] M. Kockmann, L. Burget, and J. Cernocky, "Brno university of technology system for interspeech 2010," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2822–2825, 2010.

[22] P. Nguyen, T. Le, D. Tran, X. Huang, and D. Sharma, "Fuzzy support vector machines for age and gender classification," in *Proc. 11th Annual Conference of the International Speech Communication Association*, pp. 2806–2809, 2010.

[23] T. Bocklet, A. Maier, and E. Noth, "Age determination of children in preschool and primary school age with GMM-based supervectors and support vector machines-regression," in *Proc. Text, Speech and Dialogue*, pp. 253–260, 2008.

[24] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[25] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[26] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, pp. 857–860, 2011.

[27] M. H. Bahari, M. McLaren, D. Van Leeuwen, and H. Van hamme, "Age estimation from telephone speech using i-vectors," *Proc. 13th Annual conference of the International Speech Communication Association (Interspeech)*, 2012.

[28] M. H. Bahari, N. Dehak, and H. Van hamme, "Gaussian mixture model weight supervector decomposition and adaptation," in *Internal Report*, Speech Group, 2013.

[29] M. H. Bahari, N. Dehak, and H. Van hamme, "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," in *IEEE Trans. Audio, Speech, and Lang. Process.*, Submitted, 2013.

[30] N. Dehak, O. Plchot, M. H. Bahari, and H. Van hamme, "GMM weights adaptation based on subspace approaches for speaker verification," in *SPEAKER ODYSSEY*, Submitted, 2014.

[31] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least squares support vector machines*. World Scientific, 2002.

[32] I. Fodor, "Statistical techniques to find similar objects in images," in *Proc. the American Statistical Association, Statistical Computing Section*, 2003.

[33] J. A. Suykens, L. Lukas, and J. Vandewalle, "Sparse approximation using least squares support vector machines," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 2, pp. 757–760, 2000.

[34] Y. Li, C. Lin, and W. Zhang, "Improved sparse least-squares support vector machine classifiers," *Neurocomputing*, vol. 69, no. 13, pp. 1655–1658, 2006.

[35] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 16, no. 5, pp. 980–988, 2008.

[36] M. McLaren and D. van Leeuwen, "Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 20, no. 3, pp. 755–766, 2012.

[37] M. McLaren and D. van Leeuwen, "A simple and effective speech activity detection algorithm for telephone and microphone speech," *Proc. NIST SRE Workshop*, pp. 1–6, 2011.

[38] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," pp. 213–218, 2001.

[39] X. Zhang, K. Demuynck, and H. Van hamme, "Rapid speaker adaptation in latent speaker space with non-negative matrix factorization," *Speech Communication*, 2013.

[40] M. H. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," in *Proceedings ICASSP'2013*, pp. 7344–7348, 2013.

Chapter 7

Normalized ordinal distance

This chapter is based on the following articles:
1) Bahari, M.H., Van hamme, H. (2014), "Normalized ordinal distance; a performance metric for ordinal, probabilistic-ordinal or partial-ordinal classification problems," edited book Case Studies in Intelligent Computing-Achievements and Trends, CRC Press, Taylor and Francis (Accepted).
2) Bahari, M.H., Van hamme, H., (2013), "Normalized ordinal distance; a performance metric for ordinal, probabilistic-ordinal or partial-ordinal classification problems," conf. Biometric Technologies in Forensic Science, Nijmegen, the Netherlands.

7.1 Abstract

In this chapter, a novel application-independent performance metric for ordinal, probabilistic-ordinal and partial-ordinal classification problems is introduced. Conventional performance metrics for ordinal classification problems, such as mean absolute error of consecutive integer labels and ranked probability score, are difficult to interpret and may lead to fraudulent results about the true performance of the classifier. In this chapter, first, the ordinal distance between two arbitrary vectors in Euclidean space is introduced. Then, a new performance metric, namely normalized ordinal distance, is proposed based on the introduced ordinal distance. This performance metric is conceptually simple, computationally inexpensive and application-independent. The advantages of the proposed method over the conventional approaches and its different characteristics are shown using several numerical examples.

7.2 Introduction

A large number of real world classification problems are ordinal, where there is intrinsic ordering between the categories. For example, in quality prediction systems, the task is to categorize the quality of a product into bad, good and excellent [1]. In human age group recognition from speech or images, the categories can be child, young, middle-aged and senior [2, 3]. In the classification of the therapeutic success, the classes are good recovery, moderate disability, severe disability, and fatal outcome [4]. In all ordinal classification problems (C_O), the class labels are ordinal numbers, i.e. there is intrinsic ordering between the categories.

Probabilistic-Ordinal and Partial-Ordinal Classification problems, labeled as C_O^{Pr} and C_O^{Pa} respectively, are well-known generalizations of the C_O. In C_O^{Pr}, for a test datapoint, the classifier calculates the probability of belonging to each category. In C_O^{Pa}, instead of the crisp class labels each datapoint has a degree of membership to every class [5]. These types of problems, explained in Sections 7.3.2 and 7.3.3 in detail, can be found in many domains, such as natural language processing, social network analysis, bioinformatics and agriculture [5].

Scientists have proposed different methods to solve C_O, C_O^{Pr} and C_O^{Pa} [5–10]. For example, McCullagh introduced an ordinal classifier, namely the proportional odds model (POM), based on logistic regression [6]. In [7], C_O is addressed using a generalization of support vector machines (SVM) namely support vector ordinal regression (SVOR). A neural network approach for the C_O is suggested in [8]. In [9] Gaussian processes are suggested for C_O. In [5], kernel-based proportional odds models is introduced to solve the C_O^{Pa}.

To measure the performance of these classifiers, different approaches have been suggested. For example, mean zero-one error (E_{mzo}) and mean absolute error of consecutive integer labels ($E_{\text{ma}}^{\text{cil}}$) are widely applied to measure the performance of the classifiers in C_{O} [7–10]. However, none of these methods are applicable to $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$. Percentage of correctly fuzzy classified instances (P_{cfci}) and Average Deviation (E_{ad}) have been suggested to measure the classifier performance in $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$ [5, 11–13]. The main drawback of P_{cfci} is that it does not consider the order of categories [11, 12]. The E_{ad} suggests a simple idea to solve this problem [12, 13]. Although the E_{ad} is attractive from several aspects, the interpretation of its results is difficult, because the range of its output depends on the application. The same difficulty is observed in $E_{\text{ma}}^{\text{cil}}$. Application dependency makes the interpretation of $E_{\text{ma}}^{\text{cil}}$ and E_{ad} very challenging. The average of ranked probability scores (E_{rps}), is also applied as a performance metric in $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$ [14, 15]. In this method, the order of categories is important and the range of the output is fixed between 0 and 1. This method can be applied to C_{O}, $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$. However, analysis reveals that E_{rps} over-estimates the performance of classifiers in many situations. This issue, which leads to a erroneous interpretation of classifier performance, is illustrated by some numerical examples in Section 7.6.

In this chapter, we investigate different characteristics of these performance metrics and finally a novel application-independent performance metric, namely Normalized Ordinal Distance (E_{nod}^{p}), is introduced. The Matlab code of the suggested approach, which can be applied to all three types of considered problems C_{O}, $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$, can be downloaded from our website*.

This chapter is organized as follows. In Section 7.3, the mathematical formulations of C_{O}, $\text{C}_{\text{O}}^{\text{Pr}}$ and $\text{C}_{\text{O}}^{\text{Pa}}$ are presented. In Section 7.4, five different conventional performance metrics are explained. The proposed performance metric is elaborated in Section 7.5. In Section , the effectiveness of the proposed approach is illustrated using some numerical examples. The chapter ends with a conclusion in Section 7.7.

7.3 Problem Formulation

In this section, the ordinal, probabilistic-ordinal and partial-ordinal problems are formulated.

*http://www.esat.kuleuven.be/psi/spraak/downloads/

7.3.1 Ordinal Classification

Assume that we are given a training dataset $S^{\text{tr}} = \{(X_1, Y_1), \cdots, (X_n, Y_n), \cdots, (X_N, Y_N)\}$, where $X_n = [x_{n,1}, \cdots, x_{n,i}, \cdots, x_{n,I}]$ denotes a vector of observed characteristics of the data item and $Y_n = [y_{n,1}, \cdots, y_{n,d}, \cdots, y_{n,D}]$ denotes a label vector. The label vector is defined as follows if X_n belongs to class C_d.

$$y_{n,j} = \begin{cases} 1 & j = d \\ 0 & j \neq d \end{cases} \tag{7.1}$$

In ordinal problems, there is an intrinsic ordering between the classes, which is denoted as $C_1 \prec \cdots \prec C_d \prec \cdots \prec C_D$ like low, medium and high [5]. The goal is to approximate a classifier function (G), such that for the m^{th} unseen observation X_m^{tst}, $\hat{Y}_m = G(X_m^{tst})$ is as close as possible to the true label. For a crisp classifier $\hat{Y}_m$ is defined as follows if the d^{th} class is chosen for X_m^{tst}.

$$\hat{y}_{m,j} = \begin{cases} 1 & j = d \\ 0 & j \neq d \end{cases} \tag{7.2}$$

7.3.2 Probabilistic-Ordinal classification

The probabilistic-ordinal classification problem ($\text{C}_{\text{O}}^{\text{Pr}}$) is a generalization of the C_{O}, where each element of the classifier output vector ($\hat{Y}$) represents the probability of belonging to the corresponding category. In this type of classification, Y_n is defined by relation (7.1). However, $\hat{Y}_m$ is defined as follows.

$$\hat{Y}_m = \left\{ [\hat{y}_{m,1}, \cdots, \hat{y}_{m,d}, \cdots, \hat{y}_{m,D}] \in \mathbb{R}^D \middle| \hat{y}_{m,d} \geq 0; \textstyle\sum_{d=1}^{D} \hat{y}_{m,d} = 1 \right\} \tag{7.3}$$

where $\mathbb{R}$ denotes the set of real numbers.

7.3.3 Partial-Ordinal Classification

The partial-ordinal classification problem ($\text{C}_{\text{O}}^{\text{Pa}}$) is another generalization of C_{O} [5]. In ordinal problems, each data object is limited to belong to a single category, i.e. out of all D elements of Y_n , only one is nonzero. However, this is too conservative in the case of non-crisp or fuzzy classes. This limitation is relaxed in $\text{C}_{\text{O}}^{\text{Pa}}$ by rephrasing Y_n as follows.

$$Y_n = \left\{ [y_{n,1}, \cdots, y_{n,d}, \cdots, y_{n,D}] \in \mathbb{R}^D \middle| y_{n,d} \geq 0; \textstyle\sum_{d=1}^{D} y_{n,d} = 1 \right\} \tag{7.4}$$

Therefore, each datapoint has a degree of membership to all classes. Like in ordinal problems, the final goal is to approximate a classifier function (G), such that for an unseen observation X^{tst}, $\hat{Y}_m = G(X_m^{tst})$ is as close as possible to the true label. In this type of classification $\hat{Y}_m$ is also defined by relation 7.3.

7.4 Conventional Performance Metrics

In this section, five widely-used conventional metrics, namely E_{mzo}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{ad} and E_{rps} are introduced [5–13, 15–17].

7.4.1 Mean Zero-One Error (E_{mzo})

Performance metric E_{mzo} is the fraction of incorrect predictions, which is calculated as follows [7–10].

$$E_{\text{mzo}} = \frac{1}{M} \sum_{m=1}^{M} 1_{\hat{y}_m \neq y_m} \tag{7.5}$$

where M is the total number of test set datapoints, $\hat{y}_m$ is the predicted label of the m^{th} test set datapoint and y_m is the true label of the m^{th} test set datapoint. The main advantage of E_{mzo} is its simplicity. However, it does not consider the order of the categories. Furthermore, it is not applicable to measure the performance in $\text{C}_{\text{O}}^{\text{Pr}}$ or $\text{C}_{\text{O}}^{\text{Pa}}$.

7.4.2 Mean Absolute Error of Consecutive Integer Labels ($E_{\text{ma}}^{\text{cil}}$)

To calculate the $E_{\text{ma}}^{\text{cil}}$, first, both true labels and predicted labels of the test set datapoints are transformed into consecutive integers so that if the d^{th} column of the label vector is 1 then the transformed label is equal to d [7–10]. After label transformation the $E_{\text{ma}}^{\text{cil}}$ is calculated as follows.

$$E_{\text{ma}}^{\text{cil}} = \frac{1}{M} \sum_{m=1}^{M} |\hat{U}_m - U_m| \tag{7.6}$$

where $\hat{U}_m$ is the transformed predicted label of the m^{th} test set datapoint and U_m is the transformed true label of the m^{th} test set datapoint. The $E_{\text{ma}}^{\text{cil}}$ enjoys the advantage of considering the order of categories into account. However, it cannot be applied to evaluate the classifiers in $\text{C}_{\text{O}}^{\text{Pr}}$ or $\text{C}_{\text{O}}^{\text{Pa}}$ Moreover, the range of its output is application-dependent. Therefore, the interpretation of

this metric is challenging. This is shown in Section 7.6 using some numerical examples.

7.4.3 Percentage of Correctly Fuzzy Classified Instances (P_{cfci})

Performance metric P_{cfci} has been applied to measure the performance of probabilistic or fuzzy classifiers [11, 12]. It is calculated as follows:

$$P_{\text{cfci}} = \frac{100}{M} \sum_{m=1}^{M} (1 - \frac{1}{2} \sum_{d=1}^{D} |\hat{y}_{m,d} - y_{m,d}|) \tag{7.7}$$

As it can be inferred from the above relation, the order of the categories is not considered in P_{cfci}.

7.4.4 Average Deviation (E_{ad})

Performance metric E_{ad} was originally introduced by Van Broekhoven [12] to evaluate the classifiers in fuzzy ordered classification problems. It was also applied in different applications with other names [5, 13]. The E_{ad} is calculated as follows:

$$E_{\text{ad}} = \frac{1}{M} \sum_{m=1}^{M} \left\{ \sum_{d=1}^{D-1} \left| \sum_{i=1}^{d} \hat{y}_{m,i} - \sum_{i=1}^{d} y_{m,i} \right| \right\} \tag{7.8}$$

It can be interpreted from the above relation that the order of categories is important in E_{ad}. E_{ad} is also useful for classifier evaluation in C_O^{Pr} or C_O^{Pa}. However, similar to $E_{\text{ma}}^{\text{cil}}$, the range of E_{ad} is application-dependent and hence difficult to interpret.

7.4.5 Average Ranked Probability Scores (E_{rps})

The ranked probability score was originally introduced to score the output of probabilistic classifiers [14, 15]. It is defined as follows.

$$RPS_Y(\hat{Y}) = \frac{1}{D-1} \left\{ \sum_{d=1}^{D-1} \left(\sum_{i=1}^{d} \hat{y}_i - \sum_{i=1}^{d} y_i \right)^2 \right\} \tag{7.9}$$

This scoring rule can be easily extended to measure the performance of classifiers in $\mathrm{C_O}$, $\mathrm{C_O^{Pr}}$ and $\mathrm{C_O^{Pa}}$ using the following relation.

$$E_{\mathrm{rps}} = \frac{1}{M(D-1)} \sum_{m=1}^{M} \sum_{d=1}^{D-1} \left(\sum_{i=1}^{d} \hat{y}_{m,i} - \sum_{i=1}^{d} y_{m,i} \right)^2 \tag{7.10}$$

As it can be interpreted from the above relation, the order and the number of categories are important in E_{rps}. It is assumed that the maximum of the nominator of E_{rps} is $M(D-1)$. Therefore, to fix the range of E_{rps} between 0 and 1 the nominator is divided by its maximum possible value $M(D-1)$. However, this assumption is very conservative so that in many practical cases the maximum of the nominator of E_{rps} is less than $M(D-1)$. Consequently, this assumption may lead to an erroneous interpretation of the classifier performance. Numerical examples of Section 7.6 reveal this issue clearly.

7.5 Proposed Performance Metric

In this section, first, Ordinal Distance (OD) of two vectors in Euclidean space is introduced. Then, a new performance metric, namely normalized ordinal distance (E^p_{nod}), is developed based on the ordinal distance.

7.5.1 Ordinal Distance (OD)

In this section, the definition of a distance function is recaptured. Then, the Minkowski distance is described and finally, the ordinal distance is introduced as an extension of the Minkowski distance.

Distance

By definition, a distance function of two points $A = [a_1, \cdots, a_d, \cdots, a_D]$ and $B = [b_1, \cdots, b_d, \cdots, b_D]$ is a function $D : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, which satisfies the following three conditions [18]:
1. $D(A,B) \geq 0$ and $D(A,B) = 0 \Leftrightarrow A = B$
2. $D(A,B) = D(B,A)$
3. $D(A,C) \leqslant D(A,B) + D(B,C)$

A variety of distance functions have been introduced by scientists for different applications such as Minkowski distance, Mahalanobis distance, Chebyshev distance and Hamming distance [18].

The Minkowski Distance of Order p

The Minkowski distance of order p or p-norm is a distance function, which satisfies all conditions of a distance function.

$$||A - B||_p = \left(\sum_{d=1}^{D} |a_d - b_d|^p \right)^{1/p} \tag{7.11}$$

where p is a real number not less than 1. As in can be interpreted from relation (7.11), in p-norm, the order of the elements of two points A and B, is not important.

The Ordinal Distance of Order p

The notion of ordinal distance is previously used to measure the differences of two strings [19] or two histograms [20]. In this chapter, an ordinal distance of two vectors in Euclidean space is introduced. The Ordinal Distance of order p between two points A and B is defined in relation 7.12.

$$\begin{aligned} ||A - B||_p^{\mathrm{OD}} &= \left(\sum_{d=1}^{D} \left| \bar{a}_d - \bar{b}_d \right|^p \right)^{1/p} \\ \bar{a}_d &= \sum_{i=1}^{d} a_i \\ \bar{b}_d &= \sum_{i=1}^{d} b_i \end{aligned} \tag{7.12}$$

where p is a real number not less than 1. Since (7.12) is a Minkowski distance between $\bar{A} = [\bar{a}_1 \cdots \bar{a}_d \cdots \bar{a}_D]$ and $\bar{B} = [\bar{b}_1 \cdots \bar{b}_d \cdots \bar{b}_D]$, it follows that the ordinal distance of order p satisfies the conditions of Section 7.5.1.

Figure 7.1 shows the diagram of the unit circle using Minkowski and Ordinal distances of orders 1, 2 and infinity.

7.5.2 Normalized Ordinal Distance (E_{nod}^p)

In this section, a new performance metric, namely normalized ordinal distance (E_{nod}^p), is introduced to measure the performance of classifiers in $\mathrm{C_O}$, $\mathrm{C_O^{Pr}}$ and

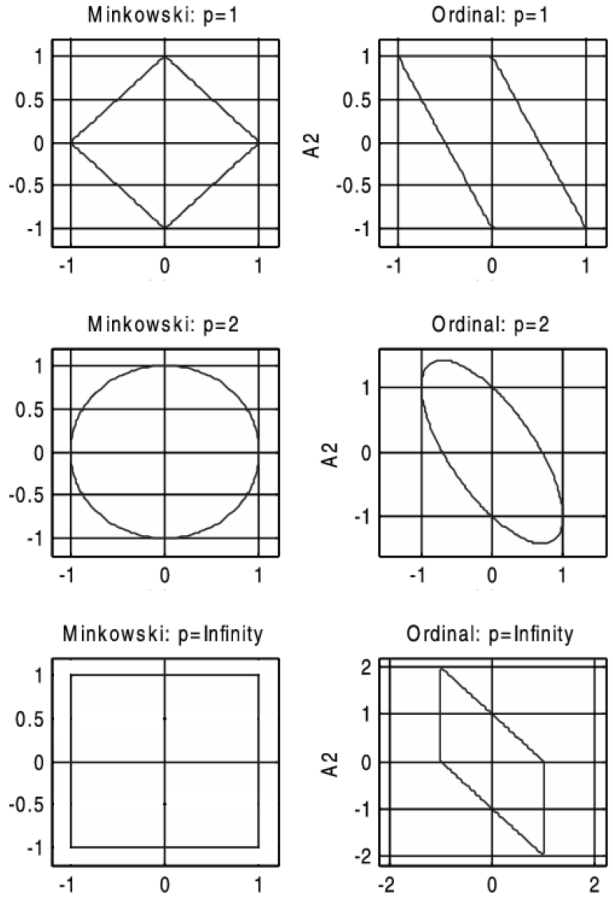

Figure 7.1: *Diagram of unit circle using Minkowski and Ordinal distances of orders 1, 2 and infinity.*

$\mathrm{C}_{\mathrm{O}}^{\mathrm{Pa}}$.

$$E_{\mathrm{nod}}^{p} = \frac{\sum_{m=1}^{M} \left\| Y_m - \hat{Y}_m \right\|_p^{\mathrm{OD}}}{\sum_{m=1}^{M} \psi_{Y_m}^{p}} \tag{7.13}$$

where $\psi_{Y_m}^{p}$ is the upper bound of $\|Y - \hat{Y}\|_p^{\mathrm{OD}}$ for any possible $\hat{Y}$ in its defined range. ψ_Y is defined as follows.

$$\psi_Y^p \triangleq \max_T \|Y - T\|_p^{\mathrm{OD}} \tag{7.14}$$

where $T = \{t_1, \cdots, t_d, \cdots, t_D\}$ is an arbitrary vector with the same specifications of $\hat{Y}$ mentioned in relation (7.2). ψ_Y^p can be calculated using theorem 1.

In E_{nod}^p ordinal distance is used to take the order of categories into account and it is normalized by the largest possible ordinal distance because not all test cases (Y_m) are equally difficult and the possible ordinal distance for some test cases is larger than others. Without this normalization the ordinal distance is difficult to interpret. In this chapter, we are performing a macro-averaging, while a micro-averaging variant could also be studied.

Theorem 1:
The upper bound of $\|Y - \hat{Y}\|_p^{\text{OD}}$ for any possible $\hat{Y}$ can be obtained as follows.

$$\psi_Y^p = max\left(\|Y - L_1\|_p^{\text{OD}}, \cdots, \|Y - L_d\|_p^{\text{OD}}, \cdots, \|Y - L_D\|_p^{\text{OD}}\right) \tag{7.15}$$

or equivalently

$$\psi_Y^p = max\left(\|Y - L_1\|_p^{\text{OD}}, \|Y - L_D\|_p^{\text{OD}}\right) \tag{7.16}$$

where L_d is a vector of size Y. The d^{th} element of L_d is equal to 1 and the rest of elements are zero. As it can be interpreted from relations (7.15) and (7.16), although the latter one is more restrictive, it provides an easier way to calculate ψ_Y^p.
Proof:
We first prove the relation (7.15), which help us to show the correctness of relation (7.16).
Proof of relation (7.15):
By definition

$$\|Y - T\|_p^{\text{OD}} = \|\Lambda(Y - T)\|_p \tag{7.17}$$

where Λ is a lower triangular matrix of size $D \times D$ with all diagonal and lower diagonal elements equal to 1. Since $\|(Y - T)\|_p$ is a convex function of T and a convex function remains convex under an affine transformation, $\|\Lambda(Y - T)\|_p$ is also convex.

On the other hand, a convex function on a compact convex set attains its maximum at an extreme point of the set [21]. In this problem $T \in \{[t_1, \cdots, t_d, \cdots, t_D] \in \mathbb{R}^D \big| t_d \geq 0; \sum_{d=1}^{D} t_d = 1\}$. The extreme points of this compact convex set are L_d with $d \in \{1, \cdots, D\}$.
Therefore

$$\max_T \|\Lambda(Y - T)\|_p = max\left(\|\Lambda(Y - L_1)\|_p^{;}, \cdots, \|\Lambda(Y - L_d)\|_p, \cdots, \|\Lambda(Y - L_D)\|_p\right) \tag{7.18}$$

Consequently

$$\max_{T} \|Y - T\|_p^{\mathrm{OD}} = max\left(\|Y - L_1\|_p^{\mathrm{OD}}, \cdots, \|Y - L_d\|_p^{\mathrm{OD}}, \cdots, \|Y - L_D\|_p^{\mathrm{OD}}\right) \tag{7.19}$$

Proof of relation (7.16):
Relation (7.16) is now shown by contradiction. Suppose relation (7.15) is not equivalent with relation (7.16), then there must be a $k \in \{2, \cdots, D-1\}$ such that

$$\|Y - L_k\|_p^{\mathrm{OD}} > \|Y - L_1\|_p^{\mathrm{OD}} \tag{7.20}$$

$$\|Y - L_k\|_p^{\mathrm{OD}} > \|Y - L_D\|_p^{\mathrm{OD}} \tag{7.21}$$

Expansion of relation (7.20) and (7.21) is

$$\sum_{d=1}^{k-1}(\sum_{i=1}^{d} y_i)^p + \sum_{d=k}^{D-1}(1 - \sum_{i=1}^{d} y_i)^p > \sum_{d=1}^{D-1}(1 - \sum_{i=1}^{d} y_i)^p \tag{7.22}$$

$$\sum_{d=1}^{k-1}(\sum_{i=1}^{d} y_i)^p + \sum_{d=k}^{D-1}(1 - \sum_{i=1}^{d} y_i)^p > \sum_{d=1}^{D-1}(\sum_{i=1}^{d} y_i)^p \tag{7.23}$$

After some manipulations (7.22) and (7.23) lead to

$$\sum_{d=1}^{k-1}\left[(\sum_{i=1}^{d} y_i)^p - (1 - \sum_{i=1}^{d} y_i)^p\right] > 0 \tag{7.24}$$

$$\sum_{d=k}^{D-1}\left[(1 - \sum_{i=1}^{d} y_i)^p - (\sum_{i=1}^{d} y_i)^p]\right] > 0 \tag{7.25}$$

If relation (7.24) holds, $(\sum_{i=1}^{d} y_i) > (1 - \sum_{i=1}^{d} y_i)$ hence $(\sum_{i=1}^{d} y_i) > 0.5$ for at least one d between 1 and $k-1$. Likewise, from (7.25), $(\sum_{i=1}^{d} y_i) < 0.5$ for at least one d between k and $D-1$. This is impossible, since $\sum_{i=1}^{d} y_i$ is an increasing function of d and hence (7.16) holds.

7.6 Results and Discussion

In this section, different characteristics of E_{nod}^{p} are discussed and its advantages to conventional performance metrics, namely E_{mzo}, P_{cfci}, E_{ad}, E_{rps}, and $E_{\mathrm{ma}}^{\mathrm{cil}}$ are demonstrated.

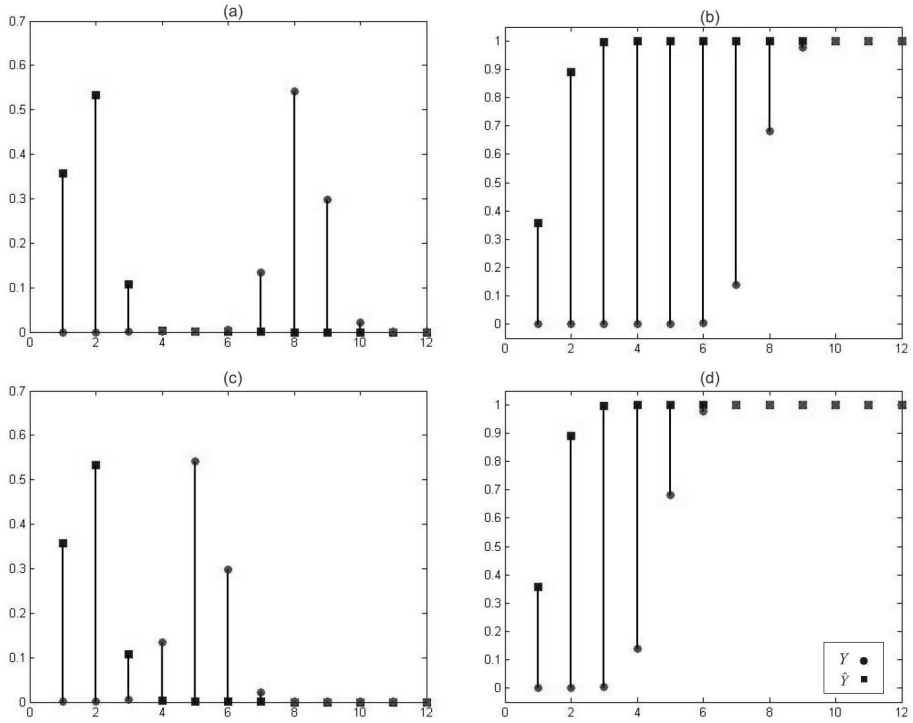

Figure 7.2: *The effect of using cumulative mass distribution.*

7.6.1 Cumulative Probability Mass Distribution

As it can be interpreted from the relation 7.13, E^p_{nod} calculates the ordinal distance between between $\hat{Y}$ and Y, which is equivalent to the Minkowski distance between cumulative probability mass distributions (CMD) of $\hat{Y}$ and Y), hence the order of categories is important. The effect of using CMD is shown in Figure 7.2 by comparing two cases. Figures 2-a and 2-b show the probability mass distributions (MD) and the CMD of $\hat{Y}$ and Y respectively for case 1. Figures 2-c and 2-d illustrate the MD and the CMD of $\hat{Y}$ and Y respectively for case 2. As it is shown in these figures, $\hat{Y}$ and Y are closer to each other in the second case compared to the first case. While the Minkowski distance between the MD of $\hat{Y}$ and Y does not reflect this fact, the Minkowski distance between CMD of $\hat{Y}$ and Y (ordinal distance of them) shows this closeness effectively.

Table 7.1: The performance of two classifiers measured by E_{mzo}, E_{ad}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞} in example 1.

Performance Metric	Problem 1	Problem 2
E_{mzo}	0.05	0.05
E_{ad}	0.05	0.1
$E_{\text{ma}}^{\text{cil}}$	0.05	0.1
P_{cfci}	97.5	97.5
E_{rps}	0.025	0.05
E_{nod}^{1}	0.0286	0.0571
E_{nod}^{2}	0.0381	0.0540
E_{nod}^{∞}	0.05	0.05

7.6.2 Order of categories

In example 1, it is shown that P_{cfci} and E_{mzo} are not suitable for measuring the performance of ordinal classifiers, because these methods do not consider the order of categories.
Example 1: For an ordinal three-class classification problem, classifier 1 and classifier 2 result in confusion matrix 1, labeled as CM_1 and CM_2 respectively. In these matrices each column represents the instances in a predicted class and each row shows the instances in an actual class.

$$\text{CM}_1 = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \text{CM}_2 = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{7.26}$$

Table 7.1 shows the performance of two classifiers measured by E_{mzo}, E_{ad}, $E_{\text{ma}}^{\text{cil}}$, P_{cfci}, E_{rps}, E_{nod}^{1}, E_{nod}^{2}, and E_{nod}^{∞}. As it can be interpreted from this table, E_{mzo}, P_{cfci} and E_{nod}^{∞} fail to reflect the degradation of performance from the classifier 1 to the classifier 2. However, E_{nod}^{1}, E_{nod}^{2}, E_{ad}, E_{rps} and $E_{\text{ma}}^{\text{cil}}$ perfectly show that classifier 1 outperforms classifier 2.

7.6.3 Number of Categories

In Example 2, it is shown that the number of categories in the classification problem influences the interpretation of E_{ad} and $E_{\text{ma}}^{\text{cil}}$.

Example 2: Consider the following three ordinal and partial ordinal classification problems.
Problem 1: For a test datapoint, the true label and the estimated label are $Y_1 = [1\ 0]$ and $\hat{Y}_1 = [0\ 1]$ respectively.
Problem 2: For a test datapoint, the true label and the estimated label are $Y_1 = [0\ 0\ 0\ 0\ 0.5\ 0.5\ 0\ 0\ 0\ 0]$ and $\hat{Y}_1 = [0\ 0\ 0.5\ 0.5\ 0\ 0\ 0\ 0\ 0\ 0]$ respectively.
Problem 3: In this problem, each two neighboring categories of Y_1 in problem 2 are merged such that the new true and estimated labels are
$Y_1 = [0\ 0\ 1\ 0\ 0]$ and $\hat{Y}_1 = [0\ 1\ 0\ 0\ 0]$ respectively.

Table 7.2 shows the performance of classifiers in these problems obtained using E_{ad}, $E^{\text{cil}}_{\text{ma}}$, E_{rps}, E^{1}_{nod}, E^{2}_{nod}, and E^{∞}_{nod} in example 2. As it can be interpreted from Table 7.2, E_{ad}, $E^{\text{cil}}_{\text{ma}}$ and E^{∞}_{nod} treated the classifiers of first and third problems in the same manner. However, the estimated label of the first problem is completely incorrect, while the estimated label in the third problem is near to the true label. Performance metrics E_{rps}, E^{1}_{nod} and E^{2}_{nod} reflect the higher performance of the third classifier compared to the first one.

The second and third problem are naturally similar to each other because the categories in the third problem are obtained by merging the neighboring categories in the second problem. An appealing characteristic of a performance metric is remaining invariant to the number of classes. It can be interpreted from Table 7.2 that the calculated performance using E_{ad} , E_{rps}, E^{1}_{nod} and E^{2}_{nod} are changed by 200%, 32%, 11% and 16% from problem 3 to problem 2. Therefore, E^{1}_{nod} and E^{2}_{nod} are robust against variability in the number of classes.

Table 7.2: The performance of two classifiers measured by E_{mzo}, E_{ad}, $E^{\text{cil}}_{\text{ma}}$, P_{cfci}, E_{rps}, E^{1}_{nod}, E^{2}_{nod}, and E^{∞}_{nod} in example 2.

Performance Metric	Problem 1	Problem 2	Problem 3
E_{ad}	1	2	1
$E^{\text{cil}}_{\text{ma}}$	1	-	1
E_{rps}	1	0.17	0.25
E^{1}_{nod}	1	0.444	0.50
E^{2}_{nod}	1	0.594	0.71
E^{∞}_{nod}	1	1	1

7.6.4 Relation to ranked probability score

There is a close relationship between E_{rps} and E^{p}_{nod}, especially for $p = 2$. In both E_{rps} and E^{p}_{nod}, denominators are assumed to be the upper bound of the numerator and are used to keep the range of performance metric between 0 and 1. In E_{rps}, it is assumed that the upper bound of the numerator is $M(D-1)$ [15, 22]. However, this is a conservative bound in many situations. In E^{p}_{nod}, this upper bound is explicitly defined by relation (7.14) and calculated by relation (7.16). The following examples show that the conservative assumption of E_{rps} results in a misleading or erroneous interpretation of the classifiers performance.

Example 3: Consider the following two cases.

Case 1:

For an ordinal three-class classification problem, a completely useless classifier is applied, which results in CM_3.

$$\text{CM}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 5 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.27}$$

Case 2:

For another ordinal three-class classification problem, consider a classifier with CM_4.

$$\text{CM}_4 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 10 & 0 & 0 \end{bmatrix} \tag{7.28}$$

The performance of classifiers in case 1 and 2 calculated by the E_{mzo}, P_{cfci}, E_{ad}, E_{rps}, E^{p}_{nod} and $E^{\text{cil}}_{\text{ma}}$ are listed in Table 7.3.

As it can be seen from Table 7.3, the performance of the applied classifier in case 1 measured by E_{rps} is 0.50, while all estimated labels are incorrect and the classifier is totally useless. The outputs of E^{p}_{nod} and P_{cfci} are 1 and 0 respectively, which appropriately reflects that the applied classifier is useless in this case. The table also indicates that E_{rps}, E_{ad} and $E^{\text{cil}}_{\text{ma}}$ result in the same values for both cases, while we know that the applied classifier in the second case is much more effective than the first one. This is appropriately reflected by E^{1}_{nod}, E^{2}_{nod} and E^{∞}_{nod}.

Example 4: This example shows the disadvantage of E_{rps} in measuring the performance of classifiers in $\text{C}^{\text{Pa}}_{\text{O}}$. Consider that in an ordinal-three-class

classification problem a probabilistic classifier is applied. The test set datapoints along with their corresponding classifier outputs are shown in Table 7.4. Performance metric E_{rps} result suggests that the classifier error is 0.2667, while it can be concluded form Table 7.4 that the applied classifier is not useful. In this example, E^1_{nod}, E^2_{nod} and E^∞_{nod} are 1, 0.73 and 0.60 respectively. Obviously, E^p_{nod} better reflects the performance of the applied probabilistic classifier especially for $p = 1$ compared to E_{rps}.

Example 5: In this example, E_{rps} and E^p_{nod} are evaluated in measuring the performance of two classifiers in a real world C_O problem, namely age group classification from speech recordings [23]. In this experiment, speech signals of 555 speakers from the N-best evaluation corpus [24] were used. The corpus contains live and broadcast commentaries, news, interviews, and reports broadcast in Belgium. The speakers of this dataset are categorized in three age categories namely, Young $(18-35)$, Middle $(36-45)$ and Senior $(46-81)$. The number of young, middle and senior speakers in this database are 138, 201 and 216 respectively. Among all speakers of the database, 400 are selected for model training and the rest are used for testing. Two approaches are applied for age group recognition. The first method is a random classifier, where $P(\hat{Y} = [1\ 0\ 0]) = P(\hat{Y} = [0\ 1\ 0]) = P(\hat{Y} = [0\ 0\ 1\ ]) = \frac{1}{3}$. The second approach, which is introduced in [23], applies well-known speech processing tools and Supervised Non-Negative Matrix Factorization (SNMF) [25] to recognize the

Table 7.3: The performance of two classifiers measured by E_{mzo}, E_{ad}, $E^{\text{cil}}_{\text{ma}}$, P_{cfci}, E_{rps}, E^1_{nod}, E^2_{nod}, and E^∞_{nod} in example 3.

Performance Metric	Case 1	Case 2
E_{mzo}	1	0.5
E_{ad}	1	1
$E^{\text{cil}}_{\text{ma}}$	1	1
P_{cfci}	0	50
E_{rps}	0.50	0.50
E^1_{nod}	1	0.5714
E^2_{nod}	1	0.5395
E^∞_{nod}	1	0.5

Table 7.4: Test set datapoints and their corresponding classifier outputs in example 4.

	Actual Label(Y)			Classifier Output($\hat{Y}$)		
Datapoint 1	0	1	0	0.3	0	0.7
Datapoint 2	0	1	0	0.6	0	0.4
Datapoint 3	0	1	0	0.5	0	0.5

age of speakers. The resulting confusion matrices of both methods can be

$$\mathrm{CM}_{\mathrm{SNMF}} = \begin{bmatrix} 15 & 15 & 9 \\ 18 & 22 & 16 \\ 9 & 11 & 40 \end{bmatrix} \mathrm{CM}_{\mathrm{random}} = \begin{bmatrix} 13 & 13 & 13 \\ 18 & 18 & 19 \\ 20 & 20 & 20 \end{bmatrix} \tag{7.29}$$

The results of using performance metrics E_{rps} and E^{p}_{nod} are listed in Table 7.5.

A subjective study on the obtained results shows that the SNMF based age group recognizer is more effective than a Random classifier. As it can be interpreted from Table 7.5, this performance drop is better revealed in E^{p}_{nod} compared to E_{rps}. In this experiment, the error of the random classifier measured by E_{rps} is only 0.44, which is not rational. By contrast, the results of E^{1}_{nod}, E^{2}_{nod} and $E^{\infty}_{\mathrm{nod}}$ effectively reflect the nature of the applied Random classifier.

7.6.5 Partial-Ordinal Problems

Examples 6 and 7 show the advantages of E^{p}_{nod} over P_{cfci}, E_{rps} and E_{ad} in measuring the performance of the classifiers in $\mathrm{C}^{\mathrm{Pa}}_{\mathrm{O}}$, where other conventional approaches are not applicable.

Example 6: In this example, P_{cfci}, E_{ad}, E_{rps}, and E^{p}_{nod} are evaluated in

Table 7.5: The performance of two classifiers measured by E_{rps}, E^{1}_{nod}, E^{2}_{nod}, and $E^{\infty}_{\mathrm{nod}}$ in example 5.

Performance Metric	SNMF	Random
E_{rps}	0.30	0.44
E^{1}_{nod}	0.37	0.54
E^{2}_{nod}	0.41	0.60
$E^{\infty}_{\mathrm{nod}}$	0.46	0.67

measuring the performance of classifiers in C_O^{Pa}. Consider an eight-class C_O^{Pa}. In this problem, the test datapoint label is $Y = [0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.2\ 0.2]$. Two classifiers are applied in this problem. Table 7.6 shows the output of the applied classifiers. The measured performance of these classifiers using P_{cfci}, E_{ad}, E_{rps}, E^1_{nod}, E^2_{nod} and E^∞_{nod} is presented in Table 7.7. As it can be understood from Table 7.6, the estimated label of the second classifier is more similar to the true label compared to that of first classifier. However, the output of the P_{cfci} is the same for both of them. This is due to the fact that the order of categories has no effect on the output of P_{cfci}. In this example, E_{ad}, E_{rps}, E^1_{nod}, E^2_{nod} and E^∞_{nod} reflect the performance improvement from the first classifier to the second one.

Example 7: In this example, the behavior of E^p_{nod} and E_{rps} in a C_O^{Pa} is analyzed. Consider a five-class C_O^{Pa}. In this problem, a special classifier is applied to recognize the labels of an infinite number of datapoints. The actual label of all datapoints is the same $Y = [0.2\ 0.2\ 0.2\ 0.2\ 0.2]$.

The applied classifier is random and crisp in which $P(\hat{Y} = [1\ 0\ 0\ 0\ 0]) = P(\hat{Y} = [0\ 1\ 0\ 0\ 0]) = P(\hat{Y} = [0\ 0\ 1\ 0\ 0]) = P(\hat{Y} = [0\ 0\ 0\ 1\ 0]) = P(\hat{Y} = [0\ 0\ 0\ 0\ 1]) = 0.2$. The error of the applied classifier expressed by the E_{rps} is 0.20. However, since the classifier is absolutely random, this result is not rational. The measured error using E^1_{nod}, E^2_{nod} and E^∞_{nod} is 0.80, 0.7983 and 0.80 respectively, which perfectly matches the characteristics of this classifier.

7.7 Conclusion

In this chapter, the ordinal distance between two arbitrary vectors in Euclidean space has been introduced. Then, Normalized Ordinal Distance (E^p_{nod}) as an application- independent performance metric for ordinal, probabilistic-ordinal or partial-ordinal classification problems has been presented. Different advantages of the E^p_{nod} over conventional performance metrics such as mean absolute error of consecutive integer labels E^{cil}_{ma}, mean zero-one error (E_{mzo}), correctly fuzzy classified instances (P_{cfci}), average deviation (E_{ad}), or ranked probability score (E_{rps}) have been shown using a number of numerical examples.

Table 7.6: The output of applied classifiers in example 6.

	Classifier Output ($\hat{Y}$)							
Classifier 1	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1
Classifier 2	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.1

Table 7.7: The performance of two classifiers measured by P_{cfci}, E_{ad}, E_{rps}, E^1_{nod}, E^2_{nod} and E^∞_{nod} in example 6.

Performance Metric	Classifier 1	Classifier 2
E_{ad}	1.2	0.2
P_{cfci}	80	80
E_{rps}	0.0314	0.0029
E^1_{nod}	0.2927	0.0488
E^2_{nod}	0.2828	0.0853
E^∞_{nod}	0.2222	0.1111

7.8 Acknowledgements

This work is supported by the European Commission as a Marie-Curie ITN-project (FP7-PEOPLE-ITN-2008), namely Bayesian Biometrics for Forensics (BBfor2), under Grant Agreement number 238803.

The authors would like to thank Prof. David van Leeuwen for helpful discussions and Dr. Jort F. Gemmeke for his help to accomplish this work.

7.9 References

[1] S. Erdural, "A method for robust design of products or processes with categorical response," *METU, Ankara*, 2006.

[2] M. H. Bahari and H. Van hamme, "Speaker age estimation and gender detection based on supervised non-negative matrix factorization," in *Proc. IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, pp. 1–6, 2011.

[3] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, no. 1, pp. 151 – 167, 2013.

[4] J. Cardoso and J. da Costa, "Learning to classify ordinal data: the data replication method," *Journal of Machine Learning Research*, vol. 8, no. 1393-1429, p. 6, 2007.

[5] J. Verwaeren, W. Waegeman, and B. De Baets, "Learning partial ordinal class memberships with kernel-based proportional odds models,"

Computational Statistics and Data Analysis, vol. 56, no. 4, pp. 928–942, 2012.

[6] P. McCullagh, "Regression models for ordinal data," *Journal of the royal statistical society. Series B (Methodological)*, pp. 109–142, 1980.

[7] W. Chu and S. Keerthi, "Support vector ordinal regression," *Neural Computation*, vol. 19, no. 3, pp. 792–815, 2007.

[8] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pp. 1279–1284, 2008.

[9] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of Machine Learning Research*, vol. 6, pp. 1019–1041, 2004.

[10] S. Shevade and W. Chu, "Minimum enclosing spheres formulations for support vector ordinal regression," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 1054–1058, IEEE, 2006.

[11] S. Manel, H. Williams, and S. Ormerod, "Evaluating presence–absence models in ecology: the need to account for prevalence," *Journal of Applied Ecology*, vol. 38, no. 5, pp. 921–931, 2002.

[12] E. Van Broekhoven, V. Adriaenssens, and B. De Baets, "Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: An ecological case study," *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 65–90, 2007.

[13] A. Mouton, B. De Baets, E. Van Broekhoven, and P. Goethals, "Prevalence-adjusted optimisation of fuzzy models for species distribution," *Ecological Modelling*, vol. 220, no. 15, pp. 1776–1786, 2009.

[14] P. Bougeault, "The wgne survey of verification methods for numerical prediction of weather elements and severe weather events," *Toulouse: Meteo-France*, 2003.

[15] A. Murphy, "On the ranked probability score," *J. Applied Meteorology*, vol. 8, pp. 988–989, 1969.

[16] J. Kohonen and J. Suomela, "Lessons learned in the challenge: making predictions and scoring them," *Lecture Notes in Artificial Intelligence*, pp. 95–116, 2005.

[17] M. Toda, "Measurement of subjective probability distributions," tech. rep., DTIC Document, 1963.

[18] M. Deza and E. Deza, *Encyclopedia of distances*. Springer, 2009.

[19] J. Morovic, J. Shaw, and P. Sun, "A fast, non-iterative and exact histogram matching algorithm," *Pattern Recognition Letters*, vol. 23, no. 1, pp. 127–135, 2002.

[20] J. Luxenburger, *Modeling and Exploiting User Search Behavior for Information Retrieval*. PhD thesis, PhD thesis, Universitat des Saarlandes, 2008.

[21] D. Kincaid and E. Cheney, *Numerical analysis: mathematics of scientific computing*, vol. 2. Amer Mathematical Society, 2002.

[22] M. Deque, J. Royer, R. Stroe, and M. France, "Formulation of gaussian probability forecasts based on model extended-range integrations," *Tellus A*, vol. 46, no. 1, pp. 52–65, 1994.

[23] M. Bahari and H. Van hamme, "Age and gender recognition from speech patterns based on supervised non-negative matrix factorization," *20th annual conference of the international association of forensic phonetics and acoustics*, pp. 3–5, 2011.

[24] D. A. van Leeuwen, J. Kessens, E. Sanders, and H. Van Den Heuvel, "Results of the n-best 2008 dutch speech recognition evaluation," *NOVA*, vol. 6, no. 10, pp. 11–5, 2009.

[25] M. Bahari and H. Van hamme, "Speaker age estimation using hidden markov model weight supervectors," in *Proc. 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 517–521, 2012.

Chapter 8

Conclusion

This chapter presents a brief overview of the thesis contributions and discusses future research directions.

8.1 Contributions

In summary the main contributions of this dissertation are:

1. A new i-vector-based approach for speaker age estimation has been proposed. This approach improves the accuracy of the conventional speaker age estimation methods significantly. This approach and obtained results have been described in Chapter 2.

2. The effect of major factors influencing the automatic age estimation systems such as utterance language and available speech duration have been investigated. The results of this investigation have been presented in Chapter 2.

3. Gaussian weight supervectors have been tested for age estimation, gender detection and accent recognition. We have reported the results of this approach in Chapter 3.

4. A Comparison of i-vectors, Gaussian mean supervectors and Gaussian weight supervectors along with the usage of different classifiers for a native-language recognition task have been performed. This comparison has been presented in Chapter 4.

5. A new subspace approach for GMM weight adaptation, namely non-negative factor analysis (NFA) have been proposed. This method applies a constrained factor analysis and suggests a new low-dimensional utterance representation approach based on Gaussian weights. This approach has been elaborated in Chapter 5.

6. It was shown that an intermediate-level fusion of the i-vector and NFA frameworks improves the recognition accuracy of the state-of-the-art i-vector-based approach in language and dialect recognition tasks. Chapter 5 reports the results of this scheme.

7. A hybrid architecture of the i-vector and the NFA frameworks for speaker age estimation has been proposed. This approach improves the state-of-the-art i-vector based system considerably. We have explained this method in Chapter 6.

8. An Ordinal Distance of two arbitrary vectors in Euclidean space was introduced. Based on the suggested distance an application independent performance metric, namely normalized ordinal distance, for ordinal, for probabilistic-ordinal and partial-ordinal classification problems has been proposed. OD and NOD can be applied in identification of many speaker characterization problems with ordinal nature such as age group recognition, identifying the level of intoxication and height group estimation.

8.2 Future Research Directions

This research has a wide range of potential extensions to increase the identification accuracy and to adapt the work to different applications in real-world scenarios.

8.2.1 Signal Representation

Development of the NFA framework as a rapid Gaussian weights adaptation approach opens new directions to improve signal representation, which is presumably the most challenging step towards developing accurate and robust speaker recognition and characterization systems.

Iterative i-vector-NFA framework

We believe that an effective approach to improve signal representation can be achieved by integrating the NFA and the i-vector frameworks. A possible combination is to extract i-vectors in two steps. In the first step, we will start by adapting the weights of the UBM to the given speech utterance. Then in the second step we will extract the i-vector based on these new weights. This new regime of extracting i-vectors can improve signal representation for speaker recognition and characterization.

Integration in GMM subspace approach

Another integration can be through coupling the i-vectors and NFA subspace vectors similar to that of the subspace GMM scheme [1], i.e. to replace the MLLR and the SMM in standard subspace GMM by the i-vector framework and the NFA method respectively.

8.2.2 NFA for phonotactic language recognition

In [2–4], SMM is applied to decompose N-gram count supervectors and reduce their dimensionality. Motivated from the success of NFA in factorizing GMM weight supervectors, we believe that NFA can replace SMM in this task to decrease the computational cost. This task may require considering multiple constraints to model conditional probabilities. These constraints can be easily imposed on the subspace matrix of the NFA framework.

8.2.3 Calibration and fusion in ordinal classification problems

Effective score calibration/fusion at the back-end of the recognition procedure plays an important role in biometrics systems, specifically in forensic applications. While calibration/fusion for speaker and language recognition is well studied, there are very few works on calibration/fusion for regression and ordinal classification problems, which are required in different speaker characterization problems such as age estimation, height recognition and identification of intoxication level. The normalized ordinal distance as a new application-independent performance metric for ordinal, probabilistic-ordinal and partial-ordinal classification problems can be applied for calibration/fusion of the output scores of ordinal classifier(s). In conventional calibration/fusion approaches, the parameters of the required mapping in calibration/fusion are optimized on a development dataset such that a conventional performance metric (e.g.

log-likelihood ratio cost) is minimized. For ordinal problems, we can determine the parameters of calibration/fusion mapping by minimizing the normalized ordinal distance instead of conventional performance metrics.

8.2.4 Adaptation to different applications

While the introduced approaches in this thesis are general, different applications may have specific requirements. For example, in forensic applications, the output of the proposed approaches should be in the form of the log-likelihood ratio. In intoxication detection in cars, considering the specific acoustic environment of the car improves the identification accuracy. In mobile phones, developing computationally inexpensive approaches to reduce the battery usage is required. Therefore, adapting the proposed approaches to specific applications plays an important role in obtaining reasonable and useful results.

8.3 References

[1] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "Subspace gaussian mixture models for speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 4330–4333, IEEE, 2010.

[2] M. Soufifar, S. Cumani, L. Burget, and J. Cernocky, "Discriminative classifiers for phonotactic language recognition with iVectors," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4853–4856, IEEE, 2012.

[3] L. F. D'haro Enrıquez, O. Glembek, O. Plchot, P. Matejka, M. Soufifar, R. d. Cordoba Herralde, and J. Cernockỳ, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts," 2012.

[4] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "iVector approach to phonotactic language recognition," in *Proc. of Interspeech*, pp. 2913–2916, 2011.

Bibliography

Mohamad Hasan Bahari received his M.Sc. degrees in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 2010, before joining the Centre for the Processing of Speech and Images (PSI), KU Leuven, Belgium, where he was granted a Marie-Curie fellowship for a PhD degree program. In February 2012, he visited the Centre for Language and Speech Technology, Radboud University Nijmegen, the Netherlands as a PhD secondment. During winter, spring and fall 2013, he visited the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), where he proposed the non-negative factor analysis (NFA) framework. His research on automatic speaker characterization was granted the Research Foundation Flanders (FWO) for a long stay abroad and awarded the International Speech Communication Association (ISCA) best student paper award at INTERSPEECH 2012. Although Mohamad Hasan's research has primarily revolved around automatic speaker characterization, his interests also extend to machine learning, and signal processing.

List of Publications

Articles in International Journals

[1] Bahari, M.H., Dehak, N., Van hamme, H., Burget, L., Ali, A., Glass, J. (2014), "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," IEEE Transactions on Audio Speech and Language Processing (Accepted).

[2] Bahari, M.H., McLaren, M., Van hamme, H., van Leeuwen D., (2014), "Speaker age estimation using i-vectors," Engineering Applications of Artificial Intelligence, Elsevier (Accepted).

[3] Bahari, M.H., Van hamme, H. (2014), "Speaker age estimation using a fusion of the i-vector and non-negative factor analysis frameworks," Pattern Recognition Letters, Elsevier (Submitted).

[4] Poorjam Alavijeh, AH., Bahari, M.H., Van hamme, H. (2014), "Automatic smoker detection from spontaneous telephone speech," Pattern Recognition Letters, Elsevier (Submitted).

Articles in Book Chapters

[1] Bahari, M.H., Van hamme, H. (2014), "Normalized ordinal distance; a performance metric for ordinal, probabilistic-ordinal or partial-ordinal classification problems," edited book Case Studies in Intelligent Computing-Achievements and Trends, CRC Press, Taylor and Francis (Accepted).

Articles in International Conferences

[1] Bahari, M.H., Van hamme, H. (2011), "Speaker age estimation and gender detection based on supervised non-negative matrix factorization," IEEE workshop biometric measurements and systems for security and medical applications, pp. 1-6, Italy.

[2] Bahari, M.H., Van hamme, H. (2011), "Rapid speaker adaptation using maximum likelihood neural regression," IEEE Int. Conf. multimedia and expo, Spain.

[3] Bahari, M.H., Van hamme, H. (2011), "Age and gender recognition from speech patterns based on supervised non-negative matrix factorization," Annual Conf. International association of forensic phonetics and acoustics, pp. 3-5, Austria.

[4] Bahari, M.H., Van hamme, H. (2012).Speaker age estimation using hidden Markov model weight supervectors," 11^{th} Conf. information science, signal processing and their applications, pp. 517-521, Canada.

[5] Bahari, M.H., Van hamme, H. (2012).Speaker adaptation using maximum likelihood general regression," 11^{th} Conf. information science, signal processing and their applications, pp. 29-34, Canada.

[6] Bahari, M.H., McLaren, M., Van hamme, H., van Leeuwen D. (2012), "Age estimation from telephone speech using i-vectors," 13^{th} annual Conf. of the international speech communication association (INTERSPEECH), pp. 506-509, USA (**International Speech Communication Association Best Student Paper Award**).

[7] van Leeuwen, D., Bahari, M.H. (2012), "Calibration of probabilistic age recognition," 13^{th} annual Conf. of the international speech communication association (INTERSPEECH), pp. 502-505, USA.

[8] Bahari, M.H., Van hamme, H., (2013), "Normalized ordinal distance; a performance metric for ordinal, probabilistic-ordinal or partial-ordinal classification problems," Conf. biometric technologies in forensic science, Nijmegen, Netherland.

[9] Bahari, M.H., Saidi, R, Van hamme, H., van Leeuwen, D. (2013), "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," International conference on acoustics, speech, and signal processing (ICASSP), pp. 7344-7348, Canada.

[10] Dehak, N., Plchot, O., Bahari, M.H., Burget, L., Van hamme, H., Dehak, R. (2014), "GMM weights adaptation based on subspace approaches for speaker verification," Odyssey, Finland (Accepted).

[11] Poorjam, A., Bahari, M.H., Vasilakakis, V., Van hamme, H., (2014), "Speaker height estimation from spontaneous telephone speech using i-Vectors," International conference on telecommunications and signal processing, Germany (Accepted).

Technical Reports

[1] Bahari, M.H., Dehak, N., Van hamme, H. (2013), "Gaussian mixture model weight supervector decomposition and adaptation," Technical report KUL/ESAT /PSI/1302, KU Leuven, ESAT, Belgium.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
CENTER FOR PROCESSING SPEECH AND IMAGES (PSI)
Kasteelpark Arenberg 10 - box 2441
B-3001 Heverlee
mohamadhasan.bahari@esat.kuleuven.be
http://www.kuleuven.be/wieiswie/en/person/00072245