HAMID BEHRAVAN

Advances in Automatic Foreign Accent Recognition

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 241

Academic Dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in Louhela auditorium in Science Park building at the University of Eastern Finland, Joensuu, on November 28, 2016,
at 12 o'clock noon.

School of Computing

Grano Oy
Jyväskylä, 2016
Editor: Prof. Pekka Toivanen

Distribution:
University of Eastern Finland Library / Sales of publications
P.O. Box 107, FI-80101 Joensuu, Finland
http://www.uef.fi/kirjasto

ISBN: 978-952-61-2314-1 (Print)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-2315-8 (PDF)
ISSNL: 1798-5668
ISSN: 1798-5676

Author's address: University of Eastern Finland
School of Computing
P.O.Box 111
80101 JOENSUU
FINLAND
email: behravan@cs.uef.fi / h.behravan@hotmail.com

Supervisors: Dr. Ville Hautamäki, Ph.D.
University of Eastern Finland
School of Computing
P.O.Box 111
80101 JOENSUU
FINLAND
email: ville.hautamaki@uef.fi

Dr. Tomi H. Kinnunen, Docent
University of Eastern Finland
School of Computing
P.O.Box 111
80101 JOENSUU
FINLAND
email: tomi.kinnunen@uef.fi

Reviewers: Professor Jean-François Bonastre, Ph.D.
University of Avignon
Center for Teaching and Research Interest in Computer Science
Computer Science Laboratory of Avignon
FRANCE
email: jean-francois.bonastre@univ-avignon.fr

Professor Michael Wagner, Ph.D.
Technical University of Berlin
Deutsche Telekom Laboratories
Quality and Usability Laboratory
GERMANY
email: michael.wagner@canberra.edu.au

Opponent: Professor Martti Vainio, Ph.D.
University of Helsinki
Institute of Behavioural Sciences
Speech Sciences
P.O.Box 9
00014 HELSINKI
FINLAND
email: martti.vainio@helsinki.fi

ABSTRACT

Characterizing spoken languages and foreign accents is a challenging pattern recognition task. This thesis addresses the problem of automatic foreign accent recognition, that is, the task of identifying the mother tongue (L1) of a non-native speaker given a speech utterance spoken in his or her second language (L2).

Selecting representative features which best characterize spoken foreign accents is an open and challenging research problem. This thesis proposes a hybrid approach to the problem of foreign accent recognition by combining both phonotactically-inspired and spectral approaches. In particular, primary aim of this thesis is to examine *whether it is possible to improve upon state-of-the-art spectral feature extraction methods within foreign accent recognition tasks by incorporating phonotactic knowledge.*

To this end, a universal acoustic characterization of speech utterances, originally developed for automatic speech recognition (ASR), is adopted in this thesis. A set of universal descriptors outline spoken foreign accents in terms of a common set of fundamental units, known as *speech attributes*. Speech attributes, namely manner and place of articulation, exist in all languages and the statistics of their co-occurrences can considerably differ from one language to another. In this dissertation, speech attributes are extracted and modeled in order to characterize spoken foreign accents using an *i-vector* representation paradigm. Then, a dimensionality reduction approach, based on the principal component analysis (PCA), is investigated in order to capture the temporal context of attribute feature streams. To further optimize the i-vector modeling backend for improved classification accuracy, a heteroscedastic linear discriminant analysis (HLDA) is compared and contrasted with a linear discriminant analysis (LDA).

A vast majority of the language and accent recognition systems assume a closed-set problem, where the training and the test segments correspond with one of the known target languages or accents. Practical systems, however, need to consider an open-set case

also, where the language of the test segment might not be any of the in-set languages. To this end, this work proposes an out-of-set (OOS) data selection approach in order to locate OOS data from an unlabeled development set and train an additional OOS model in the back-end.

Testing the proposed foreign accent recognition system on both the Finnish National Foreign Language Certificate (FSD) corpus and the US National Institute of Standards and Technology (NIST) 2008 speaker recognition evaluation (SRE) corpus, the experimental results indicate statistically significant improvement in foreign accent recognition accuracy, with a 45% relative reduction in average detection cost over the conventional Gaussian mixture model-universal background model (GMM-UBM) spectral-based technique. This attribute system outperforms an already excellent spectral-based i-vector system based on shifted delta cepstrum (SDC) features by 15% and 8% relative decrease in average detection cost for the Finnish and English data, respectively. Appending temporal context to the attribute feature streams yields 13% and 6% relative reduction in average detection cost over the context-independent attribute system for the Finnish and English data, respectively. The results of the open-set language identification (LID) task indicate that the proposed OOS data selection method outperforms the baseline one-class support vector machine (one-class SVM) by a 16% relative reduction in equal error rate (EER).

In summary, this dissertation advances state-of-the-art automatic foreign accent recognition by combining both phonotactically-inspired and spectral approaches. Furthermore, by incorporating the proposed OOS data selection method into modeling OOS languages, open-set LID accuracy substantially improves in comparison to using all the development set as OOS candidates.

Universal Decimal Classification: 004.934, 519.76, 801.612, 801.653

Library of Congress Subject Headings: Pattern recognition systems; Computational linguistics; Speech processing systems; Automatic speech recog-

nition; Accents and accentuation; Spectral analysis (Phonetics); Principal components analysis; Discriminant analysis

Yleinen suomalainen asiasanasto: hahmontunnistus; puheentunnistus; puhekieli; aksentti; ääntäminen; tietokonelingvistiikka

To my wife

Acknowledgments

This study was carried out during the years 2013-2016 in the School of Computing at the University of Eastern Finland. I would like to express my sincere gratitude to my supervisors, Dr. Ville Hautamäki and Dr. Tomi Kinnunen. It was my honor to be their Ph.D. student. They gave me tremendous knowledge to make my Ph.D. experience productive and inspiring. I would like to acknowledge my co-authors Professor Chin-Hui Lee, Professor Sabato Marco Siniscalchi, Dr. Tommi Kurki and Dr. Elie Khoury for their contributions to the studies in this thesis. I am specially grateful to Professor Jean-François Bonastre and Professor Michael Wagner for acting as the official pre-examiners of this thesis as well as Professor Martti Vainio for agreeing to act as the opponent of my public doctoral examination. I feel so proud that my thesis was evaluated by the leading experts in the field of speech processing.

This research has been supported by the University of Turku, the Nokia Foundation, doctoral positions and strategic funding of the University of Eastern Finland, the Academy of Finland projects 253120, 253000, and 283256, and by the Kone Foundation - Finland.

I would like to deeply thank my parents for their continuous love and supports in the worst and best moments of my life. Finally, and most importantly, I am deeply indebted to my wife, Nafiseh, for her understanding and beautiful heart. Her enormous support and encouragement are the base which my last three years has been built upon.

Kuopio, October 2016

Hamid Behravan

LIST OF ABBREVIATIONS

ANN	Artificial neural network
ASAT	Automatic speech attribute transcription
ASR	Automatic speech recognition
CMVN	Cepstral mean and variance normalization
CR	Classification rate
DCT	Discrete cosine transform
DET	Detection error tradeoff
DFT	Discrete Fourier transform
DNN	Deep neural network
ECDF	Empirical cumulative distribution function
EER	Equal error rate
EM	Expectation-maximization
FFT	Fast Fourier transform
FSD	Finnish national foreign language certificate
GMM	Gaussian mixture model
HLDA	Heteroscedastic linear discriminant analysis
HMM	Hidden Markov model
HSR	Human speech recognition
JFA	Joint factor analysis
kNN	k-nearest neighbour
KS	Kolmogorov-Smirnov
L1	Mother tongue
L2	Second language
LDA	Linear discriminant analysis
LID	Language identification
MAP	Maximum a posteriori
MFCC	Mel-frequency cepstral coefficient
ML	Maximum likelihood
MLPs	Multi-layer perceptrons
NAP	Nuisance attribute projection
NIST	National institute of standards and technology
OOS	Out-of-set
PCA	Principal component analysis
PLDA	Probabilistic linear discriminant analysis
PPRLM	Parallel phonetic recognition followed by language model
PRLM	Phonetic recognition followed by language model
RASTA	Relative spectral
SAPU	Satakunta in Speech

SDC	Shifted delta cepstrum
SVM	Support vector machine
UBM	Universal background model
WCCN	Within-class covariance normalization

LIST OF SYMBOLS

$\boldsymbol{a}$	Neural network weight matrix
b	Neural network bias term
$\boldsymbol{c}$	Cepstral coefficients
C	PCA context size
C_{avg}	Average detection cost
d	Dimensionality of feature vectors
D	Time advance in SDC computation
e	Offset term in linear SVM classifier
E	Number of cepstral coefficients
f	Frequency (Hz)
F	Neural network activation function
$\hat{\boldsymbol{F}}$	First-order Baum-Welch sufficient statistics
f_{mel}	Mel-scale frequency
G	Total number of components in a GMM
h	Decision function in neural network
k_i	Number of training feature vectors in class i
K	Total number of training feature vectors
$\text{KS}(\boldsymbol{x}_i, \boldsymbol{x}_j)$	KS statistic between feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
$\text{KSE}(\boldsymbol{x}_i)$	Average of the KS statistics for feature vector $\boldsymbol{x}_i$
l	A layer in neural network
L	Total number of layers in neural network
$\boldsymbol{m}$	A language- and channel-dependent GMM mean supervector
$\boldsymbol{m}_{\text{UBM}}$	UBM mean supervector
M	Total number of target languages
$\hat{N}$	Zeroth-order Baum-Welch sufficient statistics
$\boldsymbol{O}$	Matrix of eigenvectors
P	Time shift between consecutive blocks in SDC computation
Q	Number of blocks in SDC computation
r	Relevance factor in MAP adaptation
r_l	Number of neurons in layer l
R	Dimensionality of i-vectors
$\boldsymbol{S}$	Sequence of states in HMM
$\boldsymbol{T}$	Total variability matrix
$\boldsymbol{u}$	i-Vector
w	GMM component weight
$\boldsymbol{x}$	Feature vector
y	Language label
$\boldsymbol{z}$	Parameters of linear SVM classifier

α_j	Adaptation coefficient for mixture j
$\boldsymbol{\beta}$	Lagrange multipliers in linear SVM classifier
$\boldsymbol{\Delta}$	Delta cepstral coefficients
$\boldsymbol{\epsilon}$	Residual error term in i-vector representation
η	Learning rate in neural network
γ	Scale factor in MAP adaptation
λ	Eigenvalue
$\boldsymbol{\mu}$	d-dimensional mean vector of GMM
$\hat{\boldsymbol{\mu}}_{\text{ML}}$	Maximum likelihood estimator of $\boldsymbol{\mu}$
$\boldsymbol{\Sigma}$	Covariance matrix
$\boldsymbol{\Sigma}_b$	Between-class scatter matrix
$\boldsymbol{\Sigma}_w$	Within–class scatter matrix
$\hat{\boldsymbol{\Sigma}}_{\text{ML}}$	Maximum likelihood estimator of $\boldsymbol{\Sigma}$
ϕ	Kernel in SVM classifier
$\boldsymbol{\theta}$	GMM parameters
$\boldsymbol{\theta}_{\text{UBM}}$	UBM parameters
Θ	Probability distribution over all phone sequences
ϑ	Phone sequences

LIST OF ORIGINAL PUBLICATIONS

This thesis consists of the present review of the author's work in the field of automatic language and foreign accent recognition and the following publications by the author:

I **Hamid Behravan**, Ville Hautamäki, and Tomi Kinnunen, "Factors Affecting i-Vector Based Foreign Accent Recognition: A Case Study in Spoken Finnish," *Speech Communication* **66**, 118–129 (2015).

II **Hamid Behravan**, Ville Hautamäki, Sabato Marco Siniscalchi, Tomi Kinnunen, and Chin-Hui Lee, "Introducing Attribute Features to Foreign Accent Recognition," in Proc. of *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5332–5336, Florence, Italy, 2014.

III **Hamid Behravan**, Ville Hautamäki, Sabato Marco Siniscalchi, Elie Khoury, Tommi Kurki, Tomi Kinnunen, and Chin-Hui Lee, "Dialect Levelling in Finnish: A Universal Speech Attribute Approach," in Proc. of *INTERSPEECH*, pp. 2165–2169, Singapore, 2014.

IV **Hamid Behravan**, Ville Hautamäki, Sabato Marco Siniscalchi, Tomi Kinnunen, and Chin-Hui Lee, "i-Vector Modeling of Speech Attributes for Automatic Foreign Accent Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**, 29–41 (2016).

V **Hamid Behravan**, Tomi Kinnunen, and Ville Hautamäki, "Out-of-Set i-Vector Selection for Open-set Language Identification," in Proc. of *The Speaker and Language Recognition Workshop (Odyssey)* (2016), pp. 303–310, Bilbao, Spain, 2016.

Throughout the overview part of the thesis, these papers will be referred to by Roman numerals. The papers are included at the end of the thesis by the permission of their copyright holders.

AUTHOR'S CONTRIBUTION

In the first publication [**I**], the author performed all of the experiments and wrote the paper. The idea for metadata analysis of foreign accented speech data was formed by discussion between the author and his fellow co-authors, whose primary role was to supervise the work. In [**II**], the attribute-based characterization of a speech signal was adopted from the earlier work of Prof. Lee and Prof. Siniscalchi and their colleagues. The author contributed by adopting this method into the problem of foreign accent recognition, by performing a majority of the experiments and writing the paper. The idea to use PCA to capture temporal context was taken from Dr. Kinnunen. In [**III**], the SAPU (Satakunta in Speech) corpus was provided by Dr. Kurki. Further, he contributed to the leveling analysis. In this paper, the speaker diarization implementation was contributed by Dr. Khoury and the neural network attribute detectors were contributed by Prof. Siniscalchi. The author was responsible for conducting the remaining experiments and writing the paper. In [**IV**], the author worked together with the other co-authors in writing the paper. He performed all of the experiments, except for the neural network attribute detectors that were contributed by Dr. Siniscalchi. In the final publication [**V**], the author proposed an approach to the problem of OOS data selection within the context of an open-set LID. The author independently developed the methods, performed all of the experiments and wrote the paper. Protocol and data split design were suggested by the co-authors. In all of the papers, the co-authors assisted by refining the text.

Contents

1 Introduction

Speech contains a number of information cues [1], including the spoken message content [2] and speaker-related information such as identity [3], language [4], age [5], gender [6], regional origin [**III**] and emotion [7].

Language defines a set of rules for generating speech [8]. According to the most recent edition of *Ethnologue* [9], over 7,000 known living spoken languages exist in the world. Most languages belong to a certain *language family*. A family defines a group of related spoken languages that share a common ancestor. For example, Italian and Spanish both descend from Latin. Languages within a family reflect similarities in grammar and lexicon, i.e. by exhibiting similar sounds and meaning [10]. Individual languages may also have variations, known as *dialects* [11]. When a dialect is associated with a geographical region, it is called a *regional* dialect [11]. Individuals living in the same geographical region often share regional speech patterns that differ from that of individuals in other regions. *Social* dialect [11] refers to a variant of a language spoken by a social community, such as an ethnic group or age group in a particular social situation.

Another type of language variation occurs when non-native speakers use the characteristics of their mother tongue (L1) in a second language (L2) [12]. This type of variation is known as a *foreign accent* [11]. For example, when non-native speakers are learning specific sound patterns in L2 which do not exist in their L1, they tend to substitute them with the closest sound patterns in their L1 [13]. These new sounds seem *foreign* or *wrong* to the native speakers of that language [13]. For example, a German speaker might have problems pronouncing the consonants at the beginning of the English words *this* and *wish* because they do not exist in German [14]. Instead, they may pronounce them as /z/ and /v/ respectively, since they are the closest sounds found in German [14].

Foreign accent recognition refers to an automatic process of identifying the foreign accent spoken in a given speech sample. It is an enabling technology in a large number of speech processing applications. It is used in developing computer-based systems for automatically grading the pronunciation quality of students learning a foreign language [15, 16]. Non-native accents can also be found in targeted advertisements. In this form of advertisement which is based on customer's traits, connecting the customer to an agent with a similar foreign accent facilitates communication and creates a user-friendly environment [17]. In the areas of intelligence and security, automatic foreign accent recognition could help officials discover the true origin of travelers and immigrants by detecting their L1 from their speech samples [18].

Currently, ASR [19] systems are widely used in the market by internet applications. Many of these systems have been developed for the *standard* accent of a language, thus their performance degrades considerably when faced with non-native speech [20–23]. According to [24], gender and accent are the first two principal components of variation between speakers. Foreign accented speech causes a shift within the acoustic-phonetic feature space of speech [2]. This shift may considerably vary based on the speaker's proficiency in their L2 [2] and their educational background [**I**]. Finding an effective foreign accent compensation technique remains one of the most challenging problems associated with different ASR tasks [20, 21].

To deal with the problem of foreign accented speech, typical ASR systems use acoustic models adapted from foreign accented speech data [22, 25]. A relative reduction of 23.9% in error rate over a baseline native system consisting of a 3-state hidden Markov model (HMM) [26] with 48 Gaussians per each state, was observed in [22] by employing a *maximum a posteriori* (MAP) adaption [27] of acoustic models with non-native speech data. The authors in [25] proposed an adaptation approach that takes advantage of accent detection, accent discriminative acoustic features and acoustic adaptation for accented Chinese speech recognition. They achieved a 1.4% absolute reduction in character error rate when the degree

of accent considerably varied. Such acoustic adaptations improve speech recognition performance, specifically when the foreign accent is strong [20]. However, as presented in [21], using strong foreign accents leads to a lower performance of native speech recognition. To deal with this problem, authors in [20] established a systematic use of a foreign accent adapted model based on the automatic classification of speakers according to their degree of foreign accent. This approach avoided degradation of recognition performance on native speech while improving the recognition of foreign accented speech.

Recently, with the rise of *deep neural network* (DNN) acoustic modeling approaches [28, 29], the foreign accented ASR technology has led to impressive improvements, enabled by DNN capabilities, in automatically learning feature representations [29]. A multi-accent DNN with an accent-specific top layer and shared bottom hidden layers, was proposed in [28]. The accent-specific top layer was used to model distinct accent classes and the shared hidden layers allowed maximum data sharing and knowledge transfer between accented and native speech.

The selection of features, which highlight representative aspects of the speech signal for a given task, is an established problem within automatic speech processing. There has been a massive amount of studies done, since the development of digital computers beginning with the *Fourier transform*, which has given rise to a number of successful feature extraction techniques, such as the *mel-frequency cepstral coefficients* (MFCCs) [30]. The MFCCs are purely *acoustic features*, i.e. accounting for physical sound patterns. Studies have shown that *acoustic* and *phonotactic* features are the most effective language cues [4]. However, it is still not well-understood how best to characterize *foreign accent* variation. The author of this thesis explores a number of state-of-the-art foreign accent recognition approaches by adopting similar techniques from language recognition. Specifically, a new type of architecture is developed in this dissertation. Using the developed system, the author attempts to answer the following research questions:

- **Q1**: Can we improve upon the state-of-the-art spectral feature extraction methods by incorporating phonotactically-inspired knowledge to a spectral-based foreign accent recognition system?
- **Q2**: How do speaker-related characteristics such as age, education, L2 proficiency and region of origin affect foreign accent recognition accuracy?
- **Q3**: With respect to limitation of foreign accent corpora and given that training state-of-the-art foreign accent classifiers often requires vast amounts of offline speech data for training the various system components, what data can be used for training hyper-parameters of the foreign accent recognition system?
- **Q4**: Can we improve upon state-of-the-art SDC features by using alternative temporal context modeling of speech attribute features?
- **Q5**: How can one automatically select the most representative OOS data to model OOS languages from a large set of unlabeled data in order to improve the recognition accuracy of an open-set LID system?
- **Q6**: How does training set size and test utterance length affect foreign accent recognition performance overall?

To answer these research questions, the acoustic characteristics of spoken foreign accents of two languages, *English* and *Finnish*, are modeled with the help of statistical pattern recognition techniques. Then, an unknown foreign accented speech is evaluated against all available models and the most likely foreign accented model is then selected.

Particularly, to address **Q1**, the usefulness of conventional and proposed acoustic-phonetic feature vectors are compared and contrasted with respect to foreign accent recognition results. In order to discover the impact of speaker-related characteristics on foreign

accent recognition in **Q2**, speakers are first divided into meaningful subgroups within each characteristic. Then, foreign accent recognition accuracy within each subgroup is compared. As an example, speakers are divided into subgroups of elementary, high school, vocational, polytechnic and university in order to analyze the impact of *education* on foreign accent recognition results. Since training state-of-the-art foreign accent classifiers (or generally current machine learning systems in this field) often demands vast amounts of offline speech data for training the various system components [31], the effect of incorporating a non-matched dataset on foreign accent recognition results is explored in **Q3**. To address **Q4**, a simple feature stacking approach to capture temporal information from a longer context is proposed and compared against "state-of-the-art", i.e. SDC features, with comparable settings. For **Q5**, an efficient OOS data selection method is proposed and compared with several conventional OOS selection methods. Then, the proposed method is integrated into an open-set LID task to represent the OOS classes in the back-end. Finally, the recognition error rates as a function of training set size and test utterance length, are studied separately in **Q6**. In particular, for the effect of training set size, the training data is split into portions of 20%, 40%, 60%, 80% and 100% of the entire training material so that each individual portion contains the data from the previous portion. Similarly, for the effect of test utterance length, feature vectors are extracted from the 0%, 40%, 60%, 80% and 100% portions of active speech frames for evaluation.

The rest of this thesis is organized as follows. In Chapter 2, fundamentals of foreign accent recognition, including acoustic and phonotactic approaches, are presented. Open-set LID and the proposed OOS data selection method are discussed in Chapter 3. Next, in Chapter 4, the i-vector representation of spectral features and conventional channel compensation techniques in this paradigm are reviewed. Chapter 5 describes the proposed foreign accent recognition approach in detail. Chapter 6 describes standard protocols for performance evaluation in language and foreign accent recognition tasks. A summary of the publications and selected re-

sults from the publications are given in Chapter 7. Also, the proposed foreign accent recognition results are compared with other results obtained from the literature in Chapter 7. Finally, conclusions are drawn and future works are outlined in Chapter 8. The original research papers, that address the stated research questions, are attached at the end of this thesis.

2 Fundamentals of automatic foreign accent recognition

Automatic foreign accent recognition [32,33] refers to the task of modeling and classifying the foreign accent spoken in a given speech sample. Foreign-accented speech typically contains information regarding the speaker's linguistic origin, i.e. his or her native language [34]. Techniques to perform foreign accent recognition are typically adopted from LID. Language identification is the process of automatically recognizing the language of a spoken utterance [35]. Foreign accent recognition can be viewed as a specific type of LID task, where the goal is to recognize the foreign accent spoken in a given utterance. Hence, it is a common practice to adopt LID techniques to foreign accent recognition tasks.

Language identification systems operate in two phases: training and testing. In the training phase, language-dependent characteristics of the training data are modeled. During the testing phase, feature vectors are computed from a new utterance and compared to each of the language-dependent models in order to produce a set of detection scores. The language with the highest detection score is then hypothesized as the spoken language.

Figure 2.1 illustrates two broad categories of LID techniques, phonotactic [36,37] and spectral [38,39]. Phonotactic approaches typically employ phone recognizer outputs, such as N-gram statistics, to build N-gram language models. Spectral approaches in turn classify languages using the acoustic characteristics of the speech signals followed by bag-of-speech-frame models such as Gaussian mixture models (GMMs) [40].

In this Chapter, we first review acoustic-phonetic underpinnings of foreign-accented speech. Then, we describe the fundamental techniques of automatic language and foreign accent recognition,

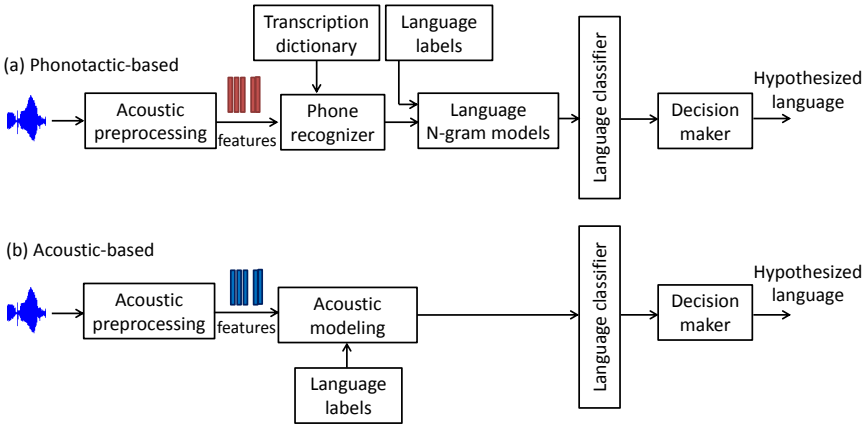


Figure 2.1: Block diagram of two commonly used approaches in automatic language recognition systems [4]: a) Phonotactic-based and b) Acoustic based techniques.

including both spectral and phonotactic approaches as well as describing the acoustic and phonotactic feature extraction process. For each approach, the possible choices for classifiers are presented.

2.1 ACOUSTIC-PHONETIC UNDERPINNINGS OF FOREIGN-ACCENTED SPEECH

Foreign-accented speech contains a number of cues about the mother tongue of the L2 speaker. Foreign accent recognition systems use one or more of these cues in order to classify accents [33]. Studies on the perception of foreign-accented speech by native speakers suggest that segmental cues, i.e. *vowels* and *consonants*, play an important role in recognizing foreign-accented speech [34, 41]. For example, Japanese speakers may not distinguish between the English consonants /l/ and /r/ as they have no English-type /l/ or /r/, but rather, their own distinct consonant which lies in between the two sounds [42]. As a result, they may mistakenly switch /l/ and /r/ in the English words *right* and *light*. Japanese speakers may also have difficulties in pronouncing a group of consecutive

consonants in English, such as /spl/ and /ŋθs/. In Japanese, a syllable contains either a single vowel or one consonant followed by one vowel (consonant+vowel) or one consonant followed by /j/ and one vowel (consonant+/j/+vowel) [43]. This makes it difficult for Japanese speakers to correctly pronounce a group of English consonants. Due to this difficulty, they may pronounce a vowel sound between the consecutive consonants [43].

Foreign-accented cues can also occur at the prosodic level, which refers to a number of properties defined on the top of segments usually extended over more than one sound segment, such as syllables, words and phrases [44]. Authors in [41] showed that prosodic cues, such as *segmental duration* and *intonation*, are relevant to the perception and rating of the foreign Italian accent in English. Segmental duration defines the time period in which a given segment of speech is produced [44], while intonation is the overall melodic pattern of a sentence, i.e. the fall and rise of a voice during speech [44].

Speakers of different languages do not pronounce vowels with the same prominence [45]. *Stress* determines which syllables or vowels are more prominent (or loud) within a word [44]. Drastic difference between stress patterns of English and Chinese is one of the main reasons of pronunciation errors made by Chinese speakers of English [45]. English is a stress language, while Chinese is a tonal language. In English, stress can differentiate the meaning of words, while in Chinese, it is the tone that changes the meanings. For example, the English word *present*, depending on the position of the stress either on its first or last syllable, means *gift* (stress on first syllable) and *describe* (stress on last syllable), respectively. However, in Chinese, the majority of the syllables receive the same stress [45].

Grammar also differs from one language to another [46]. For example, Arabic does not have the verbs *'to be'* and *'to do'* [46]. Furthermore, Arabic contains only present tense, in contrast to English which has both the simple present tense and continuous form. For these reasons, Arabic speakers may produce errors when speaking English, such as *He good driver*, *When you leave home?*, *I reading a book* and *When you going?* [46].

In addition to the acoustic-phonetic influence of L1 on L2, the amount of exposure to L2 and the age of learning were also identified as strong factors on the degree of foreign accent [47]. Analysis on the perception of foreign-accented speech indicates that the degree of foreign accent is less pronounced in L2 speakers with longer residence within an L2 speaking community [47]. Also, it is suggested that younger adults typically learn L2 faster and more fluently than older ones, resulting in less degree of foreign accent in their L2 speech [47]. In [**I**], the proposed foreign accent recognition accuracy is also related to a number of affecting factors, such as, L2 language proficiency, age of entry and level of education.

2.2 SPECTRAL TECHNIQUES

Spectral approaches within language recognition tasks are based on the observation that languages differ in their sound systems [48]. In these approaches, each speech sample is represented as a sequence of short-term spectral feature vectors [4]. The feature vectors of each language are assumed to contain specific statistical characteristics specific to that language. The spectral feature vectors are used for modeling the target languages by generative or discriminative methods.

Generative methods model the joint probability distribution between observation x and class label y using parametric family of models. Then, classification is obtained by first estimating the class-conditional densities based on Bayes rule, then classifying a new data to the class with highest posterior probability. In contrast, *discriminative* classifiers directly model the class posterior probability $p(y|x)$ without assuming any prior distributions over the classes [49]. *Gaussian mixture models* [40] and *support vector machines* (SVMs) [50] are, respectively, examples of generative and discriminative classifiers used commonly in language recognition tasks. As we will discuss further in Chapter 4, the simple acoustic approaches form the basis for more advanced *factor analysis* techniques (e.g. [51]) that are the current state-of-the-art techniques in both lan-

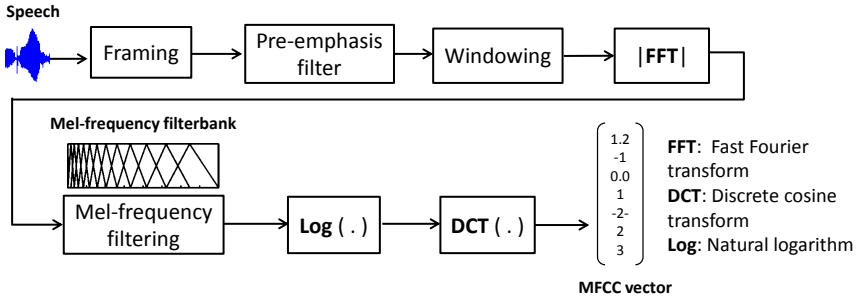


Figure 2.2: Block diagram of mel-frequency cepstral coefficient (MFCC) feature extraction [56].

guage [52] and speaker [53] recognition.

2.2.1 Acoustic feature extraction

Acoustic feature extraction refers to the process of parameterizing a raw speech signal into a sequence of feature vectors that then characterizes the information within the speech signal. According to the *source-filter* model, speaker characteristics of a speech signal are carried into the excitation source characteristics [54] and vocal tract[1] characteristics [55].

Mel-frequency cepstral coefficients [30] are commonly used for speech parameterization in a number of automatic speech processing systems. Mel-frequency cepstral coefficients are an example of speech signal parameterization using (mainly) vocal tract information. They were first introduced and applied to speech processing in [30]. Mel-frequency cepstral coefficients are loosely based on models of human auditory perception. Specifically, according to a number of psychophysical studies, human perception of the frequency content of sounds follows a nonlinear scale, known as *mel* scale. The mel scale defines a nonlinear scale of frequency based on a human's perceived frequency (or pitch).

A block diagram of MFCC computation is shown in Figure 2.2.

[1]The vocal tract is the area between the larynx and the mouth and nose where air passes during the production of sound.

In the pre-processing step, a speech sample is first divided into overlapping short speech segments or *frames* lasting 20 to 30 ms in order to obtain a local "snapshot" of signal. Next, since the high frequency components of voiced speech signals are suppressed during a human's sound production mechanism, each speech frame is filtered by a first-order *pre-emphasis filter* [57] in order to boost energy in the high-frequency region. The pre-emphasized frame is then multiplied by a tapered *window function*, usually, the so-called Hamming window [58]. The Hamming window (Figure 2.3), compared to a traditional rectangular window, results in reduced spectral leakage[2] within the spectrum of the speech frames [58]. The spectrum of each frame is obtained via *fast Fourier transform* (FFT), a computationally efficient algorithm for computing *discrete Fourier transform* (DFT) [59]. The magnitude spectrum is then passed through a bank of triangular-shaped bandpass filters positioned according to the mel-frequency scale. A commonly used analytical mapping of the frequency in Hz to mel-scale frequency in the MFCC computation is given by [60]

$$f_{\text{mel}} = 2595 \log_{10}(1 + \frac{f}{700}), \tag{2.1}$$

where f denotes the frequency in Hz and f_{mel} is the corresponding mel-scale frequency.

The mel-scale filterbank is illustrated in Figure 2.4. Here, the endpoints of each filter are positioned at the center frequencies of adjacent filters. It should be noted that MFCC implementations differ in the number of filters and the methods used to compute the filter center frequencies, among other details [60]. The number of filters, between 20 and 40, is typically selected in order to cover the signal bandwidth [0, $f_s/2$], where f_s is the sampling frequency. For a speech frame at time t, a set of E mel-scale cepstral coefficients $\boldsymbol{c}(t) = \{c_i(t), i = 0, 1, \ldots, E-1\}$, is obtained by applying a *discrete cosine transform* (DCT) [61] to the logarithm of filterbank energies so as to de-correlate them.

[2]Spectral leakage causes the spectrum of a speech frame to have new frequency components.

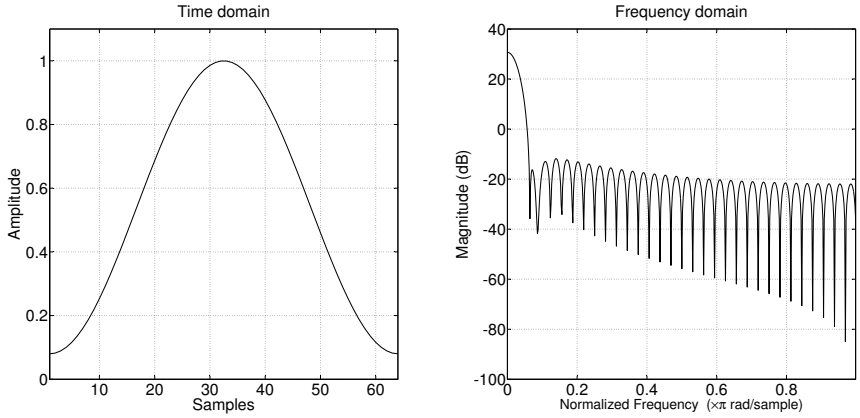


Figure 2.3: A Hamming window in both time and frequency domain.

The MFCC coefficient, c_0, indicates energy over all of the frequency bands. The second coefficient represents the balance between the high and low frequency components of the speech signal. Other coefficients do not have a clear physical interpretation, other than representing the finer details of the spectrum and allowing for discrimination between the different sounds.

The MFCC coefficients are usually further augmented with their first- and second-order derivatives, known as the *delta* and *double-delta* coefficients [62]. They are computed across several speech frames as representative of the short-term speech dynamics. A simple computation of delta coefficients $\boldsymbol{\Delta}_t$ at time t for the cepstral coefficient $\boldsymbol{c}_t$ is as follows [60]:

$$\boldsymbol{\Delta}_t = \frac{\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}}{2}, \tag{2.2}$$

where $\boldsymbol{c}_{t+1}$ and $\boldsymbol{c}_{t-1}$ are the cepstral coefficients at time $t+1$ and $t-1$, respectively.

Since the characteristics of the channel or environment might vary between different speech recordings, for example by a change in the recording microphone, it is important to reduce these unwanted channel effects on extracted feature vectors by applying

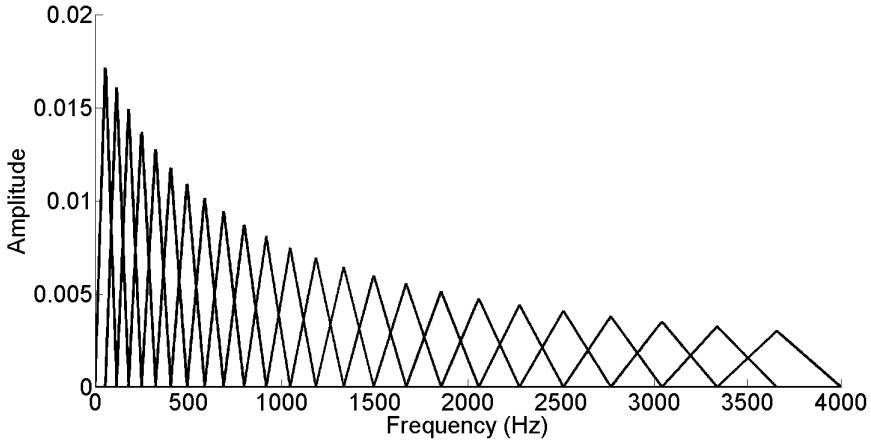


Figure 2.4: A set of 24 triangular-shaped bandpass filters for an 8 kHz sampled speech signal.

a certain type of feature compensation. The aim is to have a robust feature extraction method which is minimally affected by both channel and noise. *Relative spectral* (RASTA) filtering [63] and *cepstral mean and variance normalization* (CMVN) [64] are among the most commonly used feature compensation techniques. Cepstral mean and variance normalization normalizes the mean and variance together. After normalization, the sample mean and sample variance of each cepstral coefficient become, respectively, zero and one. RASTA filter is a band-pass filter, which can be applied in both log spectral and cepstral domain. The high-pass portion of the filter removes the effect of a channel's convolutional noise, while the low-pass portion smoothes the fast frame-to-frame spectral changes [63].

Typically, MFCC features are computed over short speech frames (e.g. 20 ms), along with their first- and second-order derivatives in order to represent temporal information. On the other hand, SDC [38] coefficients are the most commonly used set of features for language recognition tasks. Similar to the first- and the second-order delta coefficients of MFCCs, SDCs aim at capturing local speech dynamics over a longer temporal context. Shifted delta cepstrums are computed by stacking delta features of multiple speech

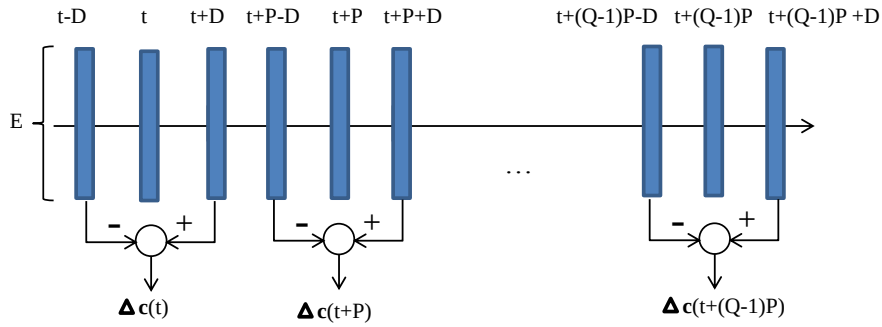


Figure 2.5: Shifted delta cepstrum feature extraction with parameters E, D, P and Q. Δ refers to the first-order derivatives of MFCCs, known as delta features.

frames. Figure 2.5 illustrates the computation of SDC features for parameters E, D, P and Q. E is the number of cepstral coefficients computed at frame t, while D is the time advance and delay for computing the first-order deltas of MFCCs. P represents the time shift between consecutive blocks and, finally, Q is the number of blocks whose delta features are concatenated to form the final SDC feature vectors. For the example displayed in Figure 2.5, the final SDC feature vector at frame t is computed through the concatenation of all of the $\Delta \boldsymbol{c}(t + iP)$:

$$\Delta \boldsymbol{c}(t) = \boldsymbol{c}(t + iP + D) - \boldsymbol{c}(t + iP - D) \tag{2.3}$$

where $i = 0, 1, \ldots, Q - 1$. Commonly used SDC parameters are $E = 7$, $D = 1$, $P = 3$, $Q = 7$. This configuration results in $E \times Q = 49$ delta features which contain temporal information of $Q \times P = 21$ consecutive frames of cepstral features.

2.2.2 Acoustic classifiers

At this point, we focus on generative and discriminative classifiers used in conventional automatic language recognition systems. Particularly, we describe GMMs as generative, and SVMs and neural networks as discriminative classifiers.

Gaussian mixture models

In language recognition systems, a GMM-UBM approach [65] is typically employed as the conventional spectral modeling technique. Gaussian mixture models have the ability to model arbitrary continuous feature distributions through the linear combination of individual Gaussian components. In language recognition, GMMs are used to model the overall characteristics of spoken languages using the distribution of acoustic feature vectors.

The d-variate Gaussian mixture densities $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, $j = 1, \ldots, G$ are given by [66]:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right]. \quad (2.4)$$

The density function of a GMM is a linear weighted combination of G Gaussian components:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{j=1}^{G} w_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \quad (2.5)$$

A GMM is denoted as parameters $\boldsymbol{\theta} = \{w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, j = 1, \ldots, G\}$, where $\boldsymbol{\mu}_j$ is a d-dimensional mean vector, $\boldsymbol{\Sigma}_j$ is a $d \times d$ covariance matrix and w_j's are the mixture weights satisfying $\sum_{j=1}^{G} w_j = 1$ and $w_j \geq 0$. Often, the choice of GMM configuration, i.e. the number of components and full or diagonal covariance matrices, is determined experimentally by the amount of data available for estimating the GMM parameters. When only a limited amount of training data is available, diagonal covariance matrices are usually adopted due to computational simplicity [66].

The parameters of a GMM, $\boldsymbol{\theta}$, are often estimated by *maximum likelihood* (ML) criterion in order to best describe the distribution of the training feature vectors. Given K training feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \ldots, K\}$, the GMM likelihood, assuming independence between the observations, can be written as:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{i=1}^{K} p(\boldsymbol{x}_i|\boldsymbol{\theta}). \quad (2.6)$$

The log likelihood form of this equation can be computed as:

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{i=1}^{K} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{K} \log \left\{ \sum_{j=1}^{G} w_j \mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\theta}_j) \right\}. \quad (2.7)$$

The ML estimation of the GMM parameters is given by:

$$\arg\max_{\boldsymbol{\theta}} \ \log\{p(\boldsymbol{X}|\boldsymbol{\theta})\}. \quad (2.8)$$

In the special case of a single Gaussian, $G = 1$, the maximum likelihood estimator of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given, respectively, by the sample mean and sample covariance matrix:

$$\hat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{x}_i \quad (2.9)$$

$$\hat{\boldsymbol{\Sigma}}_{\mathrm{ML}} = \frac{1}{K} \sum_{i=1}^{K} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathrm{ML}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathrm{ML}})^{\top}. \quad (2.10)$$

A Gaussian mixture with $G > 1$, however, (2.7) is a non-linear function of the unknown parameters $\boldsymbol{\theta}$ and a closed-form solution does not exist. An ML estimate of the GMM parameters, can however, iteratively be obtained using the so-called *expectation-maximization* (EM) algorithm [67]. An EM algorithm begins with an initial guess of the model parameters. Then at each iteration, the EM estimates the new model parameters, $\bar{\boldsymbol{\theta}}$, such that $p(\boldsymbol{X}|\boldsymbol{\theta}) \leq p(\boldsymbol{X}|\bar{\boldsymbol{\theta}})$. The new parameters then become the initial parameters for the next iteration and the process continues until convergence. The convergence can be defined, for instance, when the relative increase in log-likelihood across consecutive iterations is less than a pre-set threshold value, or when a maximum number of iterations has been reached. However, it should be noted that EM does not necessarily converge to a globally optimal ML estimate of the GMM parameters [67].

The EM algorithm to train GMMs can be summarized as follows [68]:

1. Initialize the means $\boldsymbol{\mu}_j$, covariances $\boldsymbol{\Sigma}_j$ and mixing weights w_j's, and evaluate the initial value of the log likelihood according to (2.7)

2. **E-step**: Compute the posterior probability of a mixture component j given a particular feature vector $\boldsymbol{x}_i$ and the current estimate of the parameters $\boldsymbol{\theta}$, following Bayes' rule:

$$\begin{aligned} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}) &= \frac{p(\boldsymbol{x}_i|j, \boldsymbol{\theta})p(j|\boldsymbol{\theta})}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \\ &= \frac{p(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)w_j}{p(\boldsymbol{x}_i|\boldsymbol{\theta})} \end{aligned} \tag{2.11}$$

3. **M-step**: Re-estimate the parameters using the current posterior probabilities as follows:

$$\bar{w}_j = \frac{1}{K}\sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}), \tag{2.12}$$

$$\bar{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta})\boldsymbol{x}_i}{\sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta})}, \tag{2.13}$$

$$\bar{\boldsymbol{\Sigma}}_j = \frac{\sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta})(\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\top}{\sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta})} \tag{2.14}$$

4. Evaluate the log likelihood in (2.7) according to new parameters, $\bar{w}$, $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$ from step 3. If convergence has not been reached, return to step 2.

The initial parameters of the EM algorithm can be chosen at random, for example, by selecting G random data points as the initial means and selecting the covariance matrix of the entire randomly selected data for initializing each of the G covariance matrices. Similarly, component weights can be chosen randomly, satisfying $\sum_{j=1}^{G} w_j = 1$ and $w_j \geq 0$. Other methods of initialization include using herd clustering algorithms, such as k-means in order to cluster the data first and then using the cluster means for initializing the means of the GMM components. The former approach, based on random initialization, has been adopted for the experiments in this thesis.

Maximum a posteriori adaptation of GMMs

This thesis previously described the general principle of GMMs as well as the EM algorithm. This section will now focus on the GMM-UBM [65] approach used extensively in speech classification tasks. In language recognition systems, UBM is a language-independent GMM that ideally represents the overall characteristics of the world's spoken languages. Universal background model is usually estimated by training data from all of the languages available at the time of system development. In regards to a speaker verification system, the UBM is a speaker-independent GMM trained with speech samples from a multitude of speakers representing general speech characteristics [65]. In this approach, for each hypothesized class, a GMM model is *adapted* from the UBM using a MAP estimation [27]. For classes with insufficient training data, a MAP adaptation leads to a more robust estimate of the model parameters by updating the well-trained parameters in the UBM [65]. Training a new model in the GMM-UBM approach is also faster than using the EM algorithm for ML training, requiring typically only one or two adaptation iterations.

Figure 2.6 illustrates the adaptation process of a language-specific class from UBM using new training data and UBM statistics. First, the training feature vectors are aligned with the UBM mixture components and then the adapted mixture parameters are estimated using the statistics of the training feature vectors and UBM parameters. As illustrated, different UBM mixture components are adapted for different amount depending upon the new training data.

The details of the MAP adaptation are as follows. First, given the UBM parameters $\boldsymbol{\theta}_{\text{UBM}} = \{w_j^{\text{UBM}}, \boldsymbol{\mu}_j^{\text{UBM}}, \boldsymbol{\Sigma}_j^{\text{UBM}}\}$, and K training vectors $\{\boldsymbol{x}_i, i = 1, 2, \ldots, K\}$, for each mixture j in the UBM, we first compute the posterior probabilities $p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}})$ as in (2.11). Then, new parameters are computed as follows [65]:

$$\hat{w}_j = \sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}}) \qquad \text{(weight)} \qquad (2.15)$$

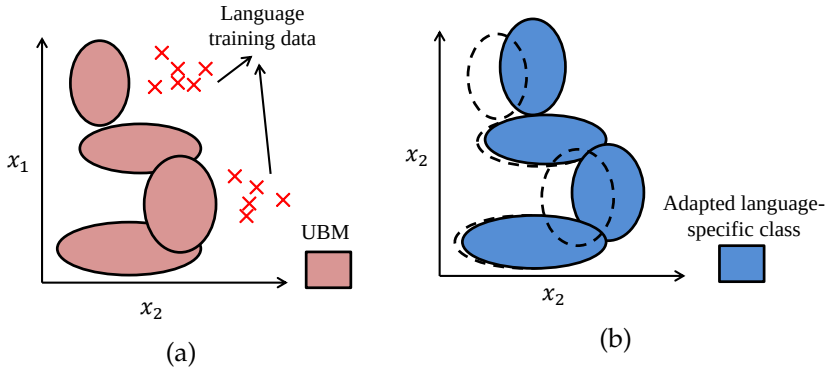


Figure 2.6: Adaptation of a target language model using the new training vectors and UBM mixture parameters. (a) The training vectors are mapped according to the UBM mixtures. (b) The adapted mixture parameters are estimated by the characteristics of the new training vectors and UBM parameters [65].

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{\hat{w}_j} \sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}}) \boldsymbol{x}_i \qquad \text{(mean)} \tag{2.16}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{1}{\hat{w}_j} \sum_{i=1}^{K} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}}) (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)^\top \quad \text{(covariance).} \tag{2.17}$$

Second, using the new parameters $\{\hat{w}_j, \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\}$ obtained from class-specific training data, the old parameters of the UBM $\{w_j^{\text{UBM}}, \boldsymbol{\mu}_j^{\text{UBM}}, \boldsymbol{\Sigma}_j^{\text{UBM}}\}$ for mixture j are adapted as follows [65]:

$$w_j^{new} = [\alpha_j^{\omega} \hat{w}_j / K + (1 - \alpha_j^{\omega}) w_j^{\text{UBM}}] \gamma \quad \text{(adapted weight)} \tag{2.18}$$

$$\boldsymbol{\mu}_j^{new} = \alpha_j^{m} \hat{\boldsymbol{\mu}}_j + (1 - \alpha_j^{m}) \boldsymbol{\mu}_j^{\text{UBM}} \qquad \text{(adapted mean)} \tag{2.19}$$

$$\boldsymbol{\Sigma}_j^{new} = \alpha_j^{v} \hat{\boldsymbol{\Sigma}}_j + (1 - \alpha_j^{v}) \boldsymbol{\Sigma}_j^{\text{UBM}} \qquad \text{(adapted covariance).} \tag{2.20}$$

For each mixture j, data-dependent adaptation coefficients $\{\alpha_j^\omega, \alpha_j^m, \alpha_j^\nu\}$ are used. These coefficients control the balance between the old and new parameters for the weights, means and variances, respectively. They are defined as [65]:

$$\alpha_j^\rho = \frac{\hat{w}_j}{\hat{w}_j + r^\rho} \,, \tag{2.21}$$

where $\rho \in \{\omega, m, \nu\}$ and r is a constant known as *relevance factor* [65]. The relevance factor is usually fixed between 6 and 16 in speaker and language recognition. During adaptation, if $\hat{w}_j$ is low for a mixture component, then $\alpha_j^\rho \to 0$ causes the old parameters to become more pronounced. Similarly, for mixture components with high $\hat{w}_j$, $\alpha_j \to 1$, emphasizes the new class-dependent parameters. The scale factor, γ, is computed over all of the adapted mixture weights to ensure their sum is equal to 1.

Support vector machines

Support vector machine (SVM) [50] is a discriminative supervised classifier. Given labeled training data, an SVM finds an optimal hyperplane (decision boundary) in order to classify training samples. For the sake of simplicity, we will consider a 2-dimensional example in Figure 2.7. In this example, we search for a decision boundary in order to linearly separate the two classes. As displayed, an infinite number of decision boundaries that separate the two classes exist. The question then, concerns which decision boundary one should use.

Intuitively, a decision boundary passing too close to the points is not an appropriate choice since it may not generalize new points correctly, such as decision boundaries 1 and 5 in Figure 2.7. Thus, an appropriate choice is to select a decision boundary that is distant from all of the other points.

The SVM is based on locating a decision boundary that is maximally far from any training data point. The minimal distance between the classes and decision boundary is known as a *margin*. The SVM is referred to as a large margin classifier since it aims at separating the training data with as much margin as possible.

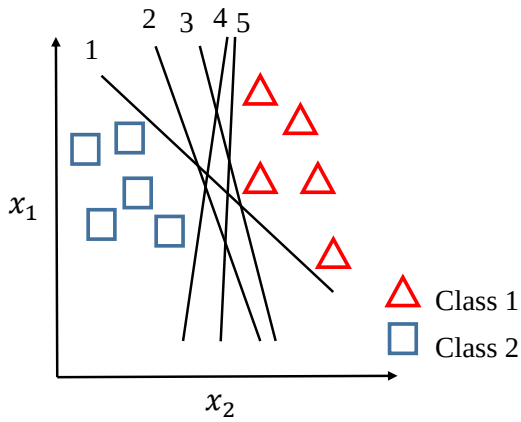


Figure 2.7: A 2-dimensional dataset with linearly separable classes.

To formulate the SVM, let us first define the equation of a hyperplane. A hyperplane is defined by $\boldsymbol{z}^\top \boldsymbol{x} + e = 0$, where $\boldsymbol{z}$ is a perpendicular vector of the hyperplane, $\boldsymbol{x}$ is a point on the hyperplane and e is a constant scalar. Changing e moves the hyperplane along the direction of $\boldsymbol{z}$, resulting in parallel hyperplanes. Mathematically, the distance between two parallel hyperplanes, $\boldsymbol{z}^\top \boldsymbol{x} + e_1 = 0$ and $\boldsymbol{z}^\top \boldsymbol{x} + e_2 = 0$, is given by [69]:

$$\frac{|e_1 - e_2|}{||\boldsymbol{z}||} \tag{2.22}$$

where $||\boldsymbol{z}||$ denotes the norm of $\boldsymbol{z}$, i.e. $||\boldsymbol{z}|| = \sqrt{z_1^2 + \cdots + z_d^2}$.

Given K training feature vectors $\{\boldsymbol{x}_i, i = 1, 2, \ldots, K\}$ and class labels $y_1, y_2, \ldots, y_K \in \{1, -1\}$, in a *linear* SVM, among a multitude choices of classifiers, the optimal hyperplane is conventionally chosen as [70]:

$$|\boldsymbol{z}^\top \boldsymbol{x} + e| = 1 \tag{2.23}$$

or equivalently:

$$\boldsymbol{z}^\top \boldsymbol{x} + e = 1 \quad \text{if} \quad y = 1, \tag{2.24}$$

$$\boldsymbol{z}^\top \boldsymbol{x} + e = -1 \quad \text{if} \quad y = -1. \tag{2.25}$$

According to (2.22), the distance between these two parallel hyperplanes is $\frac{2}{||z||}$. Since we aim to maximize the margin that separates the two classes, we need to minimize $||z||$ or, similarly, minimize $\frac{1}{2}||z||^2$.

The constraints for correct classification of all training samples are defined as: [70]

$$z^\top x_i + e \geq 1 \quad \text{if} \quad y_i = 1 \tag{2.26}$$

$$z^\top x_i + e \leq -1 \quad \text{if} \quad y_i = -1, \tag{2.27}$$

or equivalently:

$$y_i(z^\top x_i + e) \geq 1. \tag{2.28}$$

In summary, in the linear SVM classifier, for each training sample, the optimization problem is to minimize $\frac{1}{2}||z||^2$, which is subject to the inequality constraints $y_i(z^\top x_i + e) \geq 1$ for $i = 1, \ldots, K$. This is known as the *primal* formulation of linear SVMs.

In order to solve this constrained optimization problem, the Lagrange multiplier method is adopted [70]. The Lagrange multiplier method allows a *dual* form of the SVM optimization problem, written equivalently as follows [70]:

$$\Lambda(z, e, \boldsymbol{\beta}) = \frac{1}{2}||z||^2 - \sum_{i=1}^{K} \boldsymbol{\beta}_i (y_i(z^\top x_i + e) - 1), \tag{2.29}$$

where K is the number of training feature vectors and $\boldsymbol{\beta}$ is a vector of K elements with $\boldsymbol{\beta}_i \geq 0$ corresponding to the Lagrange multipliers. At the extrema, we have:

$$\frac{\partial}{\partial z}\Lambda(z, e, \boldsymbol{\beta}) = \mathbf{0}, \tag{2.30}$$

$$\frac{\partial}{\partial e}\Lambda(z, e, \boldsymbol{\beta}) = \mathbf{0}, \tag{2.31}$$

which leads to the following solutions [70]:

$$z = \sum_{i=1}^{K} \boldsymbol{\beta}_i y_i x_i, \tag{2.32}$$

$$\sum_{i=1}^{K} \boldsymbol{\beta}_i y_i = \mathbf{0}. \tag{2.33}$$

Substituting the above values into (2.29), the dual formulation of linear SVM can be rewritten as follows:

$$\Lambda(\boldsymbol{\beta}) = \sum_{i=1}^{K} \boldsymbol{\beta}_i - \frac{1}{2} \sum_{i,j=1}^{K} \boldsymbol{\beta}_i \boldsymbol{\beta}_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j. \tag{2.34}$$

This reveals that the optimization depends *only* on the inner product of pairs of feature vectors.

Given a test sample, $\boldsymbol{x}_{\text{test}}$, the decision rule in the SVM is defined as [70]:

$$\text{sign}(\boldsymbol{z}^\top \boldsymbol{x}_{\text{test}} + e). \tag{2.35}$$

Substituting $\boldsymbol{z}$ obtained from (2.32) in the decision rule leads to: [70]

$$\text{sign}(\sum_{i=1}^{K} \boldsymbol{\beta}_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}_{\text{test}} + e). \tag{2.36}$$

Again, the decision rule only depends upon the inner product between the training samples and the test feature vector.

In the development displayed above, we assumed linearly separable classes. When the data is *not* linearly separable, we can transform the data into a new space where the mapped data (features) will more likely be linearly separable.

Let ϕ define a feature mapping or *kernel* [71], which maps data to a new space. Then, to apply the SVM within the new space, rather than the original space, we map a data sample $\boldsymbol{x}$ by $\phi(\boldsymbol{x})$. Replacing $\boldsymbol{x}$ with $\phi(\boldsymbol{x})$ in (2.36) gives [70]:

$$\text{sign}(\sum_{i=1}^{K} \boldsymbol{\beta}_i y_i \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_{\text{test}}) + e). \tag{2.37}$$

A kernel is a function κ defined on feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j). \tag{2.38}$$

This indicates that in (2.37), we do not need to know ϕ explicitly. Rather, we only need to define a kernel function κ that fulfills (2.38)

for some ϕ, which itself might not be known. A number of the popular kernels with control parameters p_1, p_2 and σ^2 are:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j \qquad \text{Linear kernel}$$

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}\right) \qquad \text{Gaussian or RBF kernel}$$

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (p_1 + \boldsymbol{x}_i^\top \boldsymbol{x}_j)^{p_2} \qquad \text{Polynomial kernel}$$

The discussions above assumed that there are only two classes. In order to deal with the multi-class classification problem in language recognition or foreign accent recognition tasks for using SVMs, a "*one-versus-all*" strategy is usually adopted [72]. In this strategy, a single classifier per each class is trained considering the samples of that class as positive, and the rest of the classes as negative. After obtaining individual target class models, they are all combined to build a language recognition system.

Artificial neural networks

Artificial neural network (ANN) models historically originate from models that attempt to mimic computations carried out by the brain [73]. They were developed to simulate a network of *neurons*, which communicate with one another in the brain. An ANN implements a greatly simplified model of the actual neural computations carried out by the brain. Figure 2.8 displays a simple representation of an ANN with two inputs x_1 and x_2, and a single output (or neuron). This represents a simple type of a *single layer feed-forward neural network*. In feed-forward neural networks, each subsequent layer is connected to the previous layer without forming a cycle. In this type of ANN, the final layer produces the network's output.

The simple neuron in Figure 2.8 can be considered a computational unit which takes two inputs x_1, x_2 (and a bias term $+1$) and outputs:

$$h(\boldsymbol{x}; \boldsymbol{a}, b) = F(\boldsymbol{a}^\top \boldsymbol{x} + b) = F(\sum_{i=1}^{2} a_i x_i + b) \tag{2.39}$$

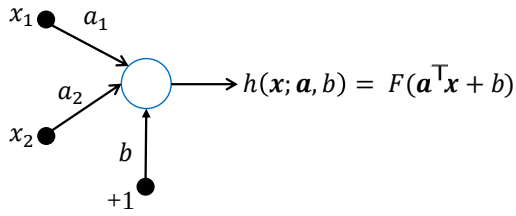


Figure 2.8: A simple neuron can be considered as a logistic unit [49] which applies a *sigmoid* (or *logistic*) function to its inputs. Here, $\boldsymbol{a} = (a_1, a_2)$ and b denote the parameters (weights) of the neural network and F denotes the activation function of the neuron.

where F is the *activation function* of the neuron, which can be either a linear or a non-linear function. In practice, the bias unit increases the capacity of the neural networks to learn nonlinear models.

Each neural network is characterized by parameters $\boldsymbol{a}$ and b. These are the connection weights between the neurons and their inputs in the entire network. Considering F as a *sigmoid* activation function[3] [76], the output (or hypothesis) is given by [77]:

$$h(\boldsymbol{x}; \boldsymbol{a}, b) = \frac{1}{1 + \exp(-\boldsymbol{a}^\top \boldsymbol{x} - b)}. \tag{2.40}$$

The aim of neural network is to produce a decision function $h(\boldsymbol{x})$ which approximates the input's target class label. The decision function $h(\boldsymbol{x})$ depends on $\boldsymbol{a}$ and b, and is therefore written as $h(\boldsymbol{x}; \boldsymbol{a}, b)$.

In general, a neural network is constructed by many of these simple neurons organized within different *layers*. Figure 2.9 displays a neural network consisting of two input units (not including the bias unit), one hidden layer with two neurons and one output layer. This neural network has parameters $\boldsymbol{a}$ and $\boldsymbol{b}$, where $a_{ij}^{(l)}$ denotes the weight associated with the weight between neuron j in layer l, and neuron i in layer $l+1$. Also, b_i^l is the bias of neuron i in layer $l+1$. According to these notations, the output (hypothe-

[3]There are also other types of activation functions such as the *hyperbolic tangent* [74] and *rectified linear function* [75].

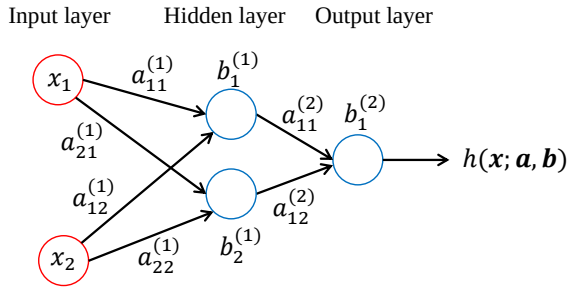


Figure 2.9: An example of a neural network with 2 input units (not including the bias unit), 1 hidden layer with 2 neurons, and 1 output layer.

sis) of the neural network in Figure 2.9 can be computed through a process called *forward propagation*, as follows:

$$h(\boldsymbol{x};\boldsymbol{a},\boldsymbol{b}) = F(a_{11}^{(2)}F(x_1a_{11}^{(1)} + x_2a_{12}^{(1)} + b_1^{(1)}) + a_{12}^{(2)}F(x_1a_{21}^{(1)} + x_2a_{22}^{(1)} + b_2^{(1)}) + b_1^{(2)}). \quad (2.41)$$

Parameters of the neural network are trained using a set of training samples. *Backpropagation* [77] is one of the most commonly used supervised learning algorithms for training neural networks. It also searches for the minimum of a defined error (or loss) function, with respect to weights, using a gradient descent method. One of the most commonly used error functions is the squared error [77]:

$$\sum_{i=1}^{K}(h(x_i;\boldsymbol{a},\boldsymbol{b}) - y^{(i)})^2, \quad (2.42)$$

where K denotes the number of training samples and y is the training sample label. The set of weights which minimize this error function are considered the solution to the training problem.

Following the notations discussed above, the decision function

$h(\boldsymbol{x})$ can be recursively computed as [78]:

$$\begin{aligned} h^{(1)} =& \boldsymbol{x} \\ h^{(2)} =& F((\boldsymbol{a}^{(1)})^\top h^{(1)} + \boldsymbol{b}^{(1)}) \\ & \dots \\ h^{(L-1)} =& F((\boldsymbol{a}^{(L-2)})^\top h^{(L-2)} + \boldsymbol{b}^{(L-2)}) \\ h(\boldsymbol{x}) =& h^{(L)} = F((\boldsymbol{a}^{(L-1)})^\top h^{(L-1)} + \boldsymbol{b}^{(L-1)}) \end{aligned} \tag{2.43}$$

Note that $h^{(l-1)}$, $l = 2, \dots, L-1$ is a vector, $h(\boldsymbol{x})$ is a scalar value and L is the total number of layers. The steps of the backpropagation algorithm for estimating the neural network parameters can be summarized as[4] [78]:

1. Initialize the parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ at random
2. Feed an input $\boldsymbol{x}$ into the network
3. Perform forward propagation to compute $h^{(1)}$, $h^{(2)}$, $h^{(3)}$, ..., $h^{(L)}$
4. For the output layer, compute:

$$\delta_1^{(L)} = 2(h^{(L)} - y)F^{'}(\sum_{j=1}^{r_L-1} a_{1j}^{(L-1)} h_j^{(L-1)} + b_1^{(L-1)}), \tag{2.44}$$

   where r_L is the number of neurons in layer L and $F^{'}$ denotes the partial derivatives of the activation function F with respect to $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e. $\frac{\partial}{\partial \boldsymbol{a}} F(.)$ and $\frac{\partial}{\partial \boldsymbol{b}} F(.)$[5].

5. For each neuron, i, within the hidden layer $l = L-1, L-2, \dots, 2$, perform the *back pass* as:

$$\delta_i^{(l)} = (\sum_{j=1}^{r_l+1} a_{ji}^{(l)} \delta_j^{(l+1)}) F^{'}(\sum_{j=1}^{r_l-1} a_{ij}^{(l-1)} h_j^{(l-1)} + b_i^{(l-1)}) \tag{2.45}$$

[4]This is one particular implementation of backpropagation algorithm.

[5]The activation function F is continuously differentiable for enabling gradient-based optimization methods.

6. Compute the partial derivatives $\Delta a_{ij}^{(l)}$ and $\Delta b_i^{(l)}$ as:

$$\Delta a_{ij}^{(l)} = h_j^{(l)} \delta_i^{(l+1)} \tag{2.46}$$

$$\Delta b_i^{(l)} = \delta_i^{(l+1)} \tag{2.47}$$

7. Update $\boldsymbol{a}$ and $\boldsymbol{b}$ according to:

$$\boldsymbol{a} = \boldsymbol{a} - \eta \Delta \boldsymbol{a} \tag{2.48}$$

$$\boldsymbol{b} = \boldsymbol{b} - \eta \Delta \boldsymbol{b}, \tag{2.49}$$

   where $0 < \eta < 1$ is the *learning rate*. A large learning rate can aggressively change the parameters, while a small learning rate can result in slow convergence.

8. Iterate steps 2-7 until convergence.

Simple feed-forward neural networks are typically adopted as language back-end classifiers [79, 80]. In this dissertation, however, a simple neural network with one-hidden layer is adopted in order to generate posterior probabilities of speech attributes. These posterior probabilities are then used as feature vectors to extract i-vectors for utterance-level characterization of languages. The speech attribute extraction process will be described in detail in Chapter 5.

2.3 PHONOTACTIC TECHNIQUES

Phonotactic language recognition systems are based on co-occurrences of phone sequences in speech. Unlike spectral approaches, which extract acoustic features from fixed-length frames of speech, in phonotactic approaches, speech is segmented into a logical unit of tokens, namely, *phones*. *Phonetic recognition followed by language model* (PRLM) [81] is the basis for a majority of state-of-the-art phonotactic approaches. In PRLM, first, a phone recognizer extracts phoneme sequences from the speech data. Following this, N-gram language models are estimated by computing the probability of occurrences of all phone sequences in each target language. The N-gram models are then used to classify the phoneme sequences of the test data.

2.3.1 Phone recognition

The goal of phone recognition is to accurately recognize the phone sequences contained in speech. Accurate phone recognition has a significant impact on the accuracy of phonotactic language recognition systems. However, it also occupies a majority of the processing time in PRLM [48].

An HMM is also often adopted to perform phone recognition. It consists of a network of context-independent phones, each having three emitting states [48]. The term context-independent refers to the recognition of phones without taking the context into account [26]. Phone HMMs are trained with phonetically labeled speech data from different languages. To ensure that all of the possible sound units in the target languages are captured by the phone recognizers, it is common to train the HMM phone recognizers using data from multiple languages [82]. The primary difficulty, however, is that phonetically labeled speech data from multiple languages may not be available. Since labeled English speech is more commonly represented within the available speech corpora, a single-language phone recognizer trained on English data is typically adopted for PRLM [48].

Each phone is represented by the three *states* of HMM. Given a sequence of observations, $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \ldots, K\}$, generally in the form of acoustic feature vectors, an HMM phone recognizer decodes the most likely sequence of states $\boldsymbol{S} = \{s_i, i = 1, 2, \ldots, K\}$ which have produced the sequence of observations according to [81]:

$$\arg\max_{\boldsymbol{S}} p(\boldsymbol{S}|\boldsymbol{X}). \tag{2.50}$$

By adopting the Bayes rule and noting that maximization does not depend upon $p(\boldsymbol{X})$, (2.50) can be written as:

$$\arg\max_{\boldsymbol{S}} \frac{p(\boldsymbol{X}|\boldsymbol{S})p(\boldsymbol{S})}{p(\boldsymbol{X})} = \arg\max_{\boldsymbol{S}} p(\boldsymbol{X}|\boldsymbol{S})p(\boldsymbol{S}). \tag{2.51}$$

Eq. (2.51) defines a search space over all of the possible state sequences. It is important to note that the search space size grows

exponentially with the number of observations, making a *brute force* search infeasible for finding the most likely sequence of states (phones) [83]. Instead, an efficient algorithm known as the *Viterbi algorithm* [83] is generally adopted. This algorithm belongs to the class of *dynamic programming* algorithms that uses a table to store intermediate values as it finds the most likely sequence of states (phones).

2.3.2 Phonotactic classifier

A phone recognizer produces the most probable phone sequence, $\boldsymbol{\vartheta}$. In a PRLM, N-gram statistics [83] are employed to estimate the frequency of occurrences of phone sequences within each target language. An N-gram captures the probability of a particular phone given the $N-1$ preceding consecutive phones according to [84]:

$$p(\vartheta_i|\vartheta_{i-1}, \vartheta_{i-2}, \ldots, \vartheta_{i-(N-1)}). \tag{2.52}$$

Then, N-gram statistics are computed for all possible phoneme sequences which occur in each language to form the language model Θ. Each language model represents the phonotactic information of a particular language. Given the phone sequence $\boldsymbol{\vartheta} = \{\vartheta_1, \vartheta_2, \ldots, \vartheta_I\}$ of a particular utterance, the likelihood score for language y is obtained by [84]:

$$\ell(\boldsymbol{\vartheta}|\Theta_y) = \prod_{i=N}^{I} p(\vartheta_i|\vartheta_{i-1}, \vartheta_{i-2}, \ldots, \vartheta_{i-(N-1)}, \Theta_y). \tag{2.53}$$

where Θ_y denotes the language model corresponding to language y. Then, the most likely language, y^*, is obtained by:

$$y^* = \arg \max_{1 \leq y \leq M} \ell(\boldsymbol{\vartheta}|\Theta_y). \tag{2.54}$$

where M is the total number of target languages.

For the language recognition problem, an N-gram order is typically chosen between 2 and 4 [48]. While, higher order N-gram

counts are expected to contain more discriminant language-specific information, estimating their probabilities is difficult since the number of N-grams increases exponentially. Also, N-gram models often do not generalize from training to test set [85]. For the higher order N-grams, more training data is often needed [48].

As implied by (2.54), phone recognition and target language modeling are performed independently. PRLM can be further extended to include multiple parallel phone recognizers, each trained on different languages. The author of [86] developed multiple language-dependent phone recognizers in parallel to better capture phoneme characteristics represented in the target languages and increase the robustness of phone recognition. This system is commonly known as *parallel PRLM* (PPRLM) [86]. Parallel PRLM consists of multiple phone recognizers, each trained on different languages. These languages are not necessarily limited to the target languages. Each phone recognizer is then followed by N-gram language models trained on phoneme sequences generated from the corresponding phone recognizers. The language-specific scores obtained from each individual system are then combined using, for instance, a product-rule fusion [48] in order to produce the final detection scores.

3 Open-set versus closed-set identification

As mentioned in the previous chapter, LID is a multi-class recognition task where the objective is to assign a given input utterance to one of the language classes, $\{y_i, i = 1, \ldots, M\}$, where M is the number of target languages. In a *closed-set* scenario, each y_i represents an explicitly known or *in-set* language. In contrast, in an *open-set* scenario, speech may come from any language, either explicitly known or unknown. An open-set scenario can be addressed by simply introducing an additional OOS class into the target set [4]. Then, the objective is to assign the test segment to one of the in-set languages or a single OOS class, as illustrated in Figure 3.1.

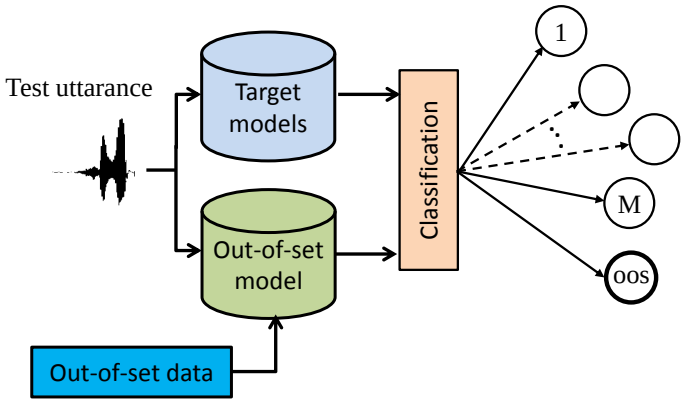


Figure 3.1: An open-set language identification (LID) can be implemented by introducing an additional out-of-set (OOS) class into the target set. The objective then, is to classify the test utterance as one of the M target languages or an OOS class.

The OOS model shown in Figure 3.1 is complementary to target languages and represents a *language-independent* distribution of fea-

tures. The OOS model can be trained using additional OOS data. A number of approaches select the OOS data based on linguistic similarities of the languages in a supervised way. Authors in [87] collected additional data from languages different from the target languages. Specifically, OOS candidates with different prosody characteristics from the target languages were selected from different language families. It should be noted that these supervised approaches require either linguistic knowledge, additional metadata or manual works, which can be costly and time consuming.

The use of unlabeled data for general system development purposes is currently being investigated within the speaker and language recognition research community. Training an OOS model from an unlabeled development set consisting of both target and OOS languages was one of the primary issues of the 2015 NIST language recognition i-vector challenge [88]. Being able to select the OOS data in an unsupervised or a *semi-automatic* manner, makes OOS modeling more practical and scalable for a larger variety of problems. For semi-automatic approaches, one might only need a small number of language labels of the reference languages for the OOS data selection to be done in an entirely unsupervised manner.

Authors in [52] proposed a best fit OOS data selection method followed by cluster purification in order to select the best OOS candidates from an unlabeled development set. First, all of the unlabeled data is assumed to be OOS. Then, a multi-class SVM is run on $M + 1$ classes (M target classes + 1 OOS). Each unlabeled data is then scored against the OOS class and those yielding the highest posterior probability are selected as the OOS candidates. Following this, an adaptive cluster purification technique is further applied in order to increase OOS homogeneity by excluding those OOS utterances which are close to the target languages.

Other classical approaches for locating OOS data from a set of unlabeled data include methods based on a one-class SVM [89] and methods based on k-nearest neighbor (kNN) distances [90]. Each of these methods provides an outlier score for each of the unlabeled data. Specifically, in a one-class SVM, the detector constructs a de-

cision boundary in order to achieve maximum separation between the training data and the origin. Then, the distance between the unlabeled data and the decision boundary is considered the outlier score. In the kNN approach, the outlier score for an unlabeled data is computed by the sum of its distances from its k nearest neighbors [90].

In [**V**], the author proposed a simple and effective OOS selection method for identifying OOS candidates from an unlabeled development set in the i-vector space [51]. The technique is based on the non-parametric *Kolmogorov-Smirnov* (KS) test [91, 92], which is used to decide whether a sample is drawn from a population with a known distribution (*one-sample* KS test) or to estimate whether two samples have the same underlying distribution (*two-sample* KS test). Using this approach, each unlabeled sample in the development set is given a per-class outlier score. Scores with a higher value confidently indicate that the corresponding sample is an OOS observation (none of the known target classes).

Particularly, for any feature vector $\boldsymbol{x}_i$, the distances of $\boldsymbol{x}_i$ to other feature vectors in language y have an empirical cumulative distribution function (ECDF) $F_{\boldsymbol{x}_i}(x)$ evaluated at points x. The KS statistic between feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in language y is defined as [93]:

$$\mathrm{KS}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_x |F_{\boldsymbol{x}_i}(x) - F_{\boldsymbol{x}_j}(x)|. \tag{3.1}$$

The outlier score for $\boldsymbol{x}_i$ is then defined as the average of the previous KS test statistics, computed as:

$$\mathrm{KSE}(\boldsymbol{x}_i) = \frac{1}{K-1} \sum_{\substack{j=1 \\ j \neq i}}^{K} \mathrm{KS}(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3.2}$$

where K is the total number of training instances in language y. The average of the KS test statistics in (3.2) lies between 0 and 1. A point is considered as an OOS to a class if its KSE is large, i.e. close to 1.

Figure 3.2 depicts an example of the distribution of in-set and OOS KSE values for a French language class using i-vector [51]

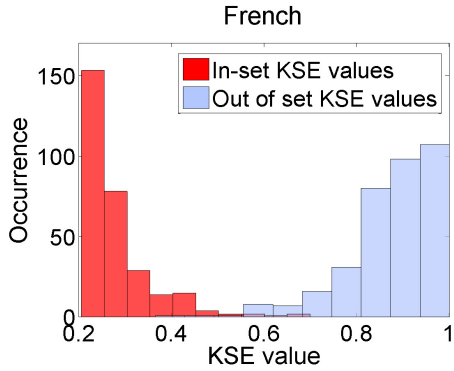


Figure 3.2: Distribution of in-set and OOS KSE values for a French language class using i-vector [51] representation of utterances (from [**V**]). The in-set KSE values tend to be close to 0, while OOS KSE values tend to be close to 1.

representation of utterances. To compute the in-set KSE values, only data from the French class was used to plot the distribution. In other words, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in (3.1) and (3.2) are both from the same class. Similarly, for OOS KSE values, a set of data which does not belong to the French class was used. In other words, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in (3.1) and (3.2) are from different classes. It is clear that the in-set KSE values tend to be close to 0, while OOS KSE values tend to be close to 1.

4 i-Vector modeling

In Chapter 2, we reviewed common choices for classifiers within the language and foreign accent recognition problems. This included both GMM-based generative approaches, as well as discriminative approaches such as SVMs and neural networks. Recent research in speaker and language recognition has focused on the *i-vector* front-end factor analysis approach. The i-vector methodology [51] was first introduced by Dehak et al. within the context of automatic speaker verification to define a new low-dimensional subspace representation of speech utterances that models both speaker and channel variabilities. The i-vector approach was obtained by modifying the existing successful supervector based *joint factor analysis* (JFA) approach [94], which models speaker and channel subspaces using separate subspace models [95–97]. One of the main differences between speaker and language recognition with respect to the JFA approach is that in the former, language is an unwanted (nuisance) variation, while in the latter, language variation is useful and comprises the desired information [98, 99].

Since the i-vector approach is based on defining *one* subspace which contains both speaker and channel variations, subsequent channel compensation techniques are required in order to suppress the effect of unwanted variability within the i-vector representation [51]. Consequently, an i-vector system can be viewed as a front-end feature extractor for further channel compensation and modeling in the back-end side [51]. Widely used channel compensation techniques for the i-vector features include *within-class covariance normalization* (WCCN) [100], *linear discriminant analysis* (LDA) [101], *probabilistic linear discriminant analysis* (PLDA) [102] and *nuisance attribute projection* (NAP) [103].

Figure 4.1 displays the block diagram of a speaker (or language) recognition system based on the i-vector representation. This diagram indicates which parts of the system are trained in supervised

and unsupervised mode. In the present context, supervised training refers to the training process of the pattern recognition system that makes no use of the available language information. While in supervised training, labels for each particular i-vector are required. In the following sections, we describe the i-vector parameter estimation and the channel compensation techniques used in this dissertation.

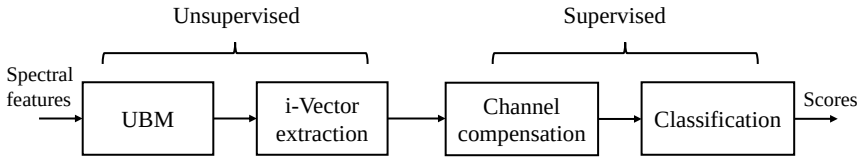


Figure 4.1: Block diagram of speaker (or language) recognition in i-vector representation paradigm indicating which parameters are trained in supervised and unsupervised mode.

4.1 THE I-VECTOR REPRESENTATION IN LANGUAGE RECOGNITION

An i-vector extractor [51] maps a sequence of acoustic feature vectors obtained from a speech utterance, $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \ldots, N\}$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, to a fixed-length vector $\boldsymbol{u} \in \mathbb{R}^R$. In order to perform this, given a UBM of G components with parameters $\boldsymbol{\theta}_{\text{UBM}} = \{w_j^{\text{UBM}}, \boldsymbol{\mu}_j^{\text{UBM}}, \boldsymbol{\Sigma}_j^{\text{UBM}}\}$, we first compute the following zeroth- and first-order centered *Baum-Welch sufficient statistics* [51]:

$$\hat{N}_j = \sum_{i=1}^{N} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}}) \tag{4.1}$$

$$\hat{\boldsymbol{F}}_j = \frac{1}{\hat{N}_j} \sum_{i=1}^{N} p(j|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{UBM}})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{\text{UBM}}). \tag{4.2}$$

Subsequently, we assume each utterance to have a language- and channel-dependent GMM mean supervector $\boldsymbol{m} \in \mathbb{R}^{G \times d}$, that obeys a *factor analysis* model according to [51]:

$$\boldsymbol{m} = \boldsymbol{m}_{\text{UBM}} + \boldsymbol{T}\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{4.3}$$

where $\boldsymbol{m}_{\text{UBM}} \in \mathbb{R}^{G\times d}$ denotes the utterance- and channel-independent component (UBM mean supervector), $\boldsymbol{T} \in \mathbb{R}^{(G\times d)\times R}$ is a global low rank *total variability* matrix where a majority of the language-specific information resides. Vector $\boldsymbol{w}$ represents a latent random variable with a prior standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and $\boldsymbol{\epsilon}$ is a residual error term with $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$, which models the variability not captured by $\boldsymbol{T}$. For a given utterance, the posterior distribution of $\boldsymbol{w}$ is computed for the entire sequence $\boldsymbol{X}$. The mean of this posterior distribution is known as the i-vector, $\boldsymbol{u}$, and is obtained as follows [51,95]:

$$\boldsymbol{u} = \left(\boldsymbol{I} + \boldsymbol{T}^{\top}\boldsymbol{\Psi}^{-1}\hat{\boldsymbol{N}}\boldsymbol{T}\right)^{-1}\boldsymbol{T}^{\top}\boldsymbol{\Psi}^{-1}\hat{\boldsymbol{N}}\hat{\boldsymbol{F}}, \tag{4.4}$$

where $\hat{\boldsymbol{N}} \in \mathbb{R}^{Gd\times Gd}$ is a diagonal matrix whose diagonal blocks are $\hat{N}_j\boldsymbol{I}$, $j = 1,\ldots,G$, and $\hat{\boldsymbol{F}} \in \mathbb{R}^{Gd\times 1}$ is a supervector generated by stacking the $\hat{\boldsymbol{F}}_j$, and $\boldsymbol{\Psi} \in \mathbb{R}^{Gd\times Gd}$ is a diagonal covariance matrix which captures the residual variability not captured by $\boldsymbol{T}$. Here, the superscript $^{\top}$ denotes a matrix transpose.

An efficient algorithm to train $\boldsymbol{T}$ is described in [95, 96]. It is estimated using the EM algorithm similar to JFA eigenvoice training [97] except that, during training all of the training utterances of a given class are treated as if they are from different classes in order to capture both language and channel variations [51].

4.2 CHANNEL COMPENSATION TECHNIQUES

As discussed earlier, in the i-vector approach, channel compensation is performed on the extracted i-vectors at the back-end. Channel compensation approaches are typically defined based on the within- and between-class variations among the i-vectors. In the context of language recognition, a within-class variation might occur due to transmission channel, acoustic environment, microphones and also intrinsic speaker variation. Similarly, a between-class variation depends on differences in language information between classes. Channel compensation techniques are typically

adopted in order to minimize the within-class variations while maximizing the between-class variations [51,95].

The dimensionality reduction of i-vectors is one of the commonly used channel compensation techniques in the i-vector approach [39,95,104,105]. These techniques attempt to reduce the dimensionality of i-vectors while preserving as much of the discriminatory information as possible for classification purposes [39,95]. Below, we review three commonly used dimensionality reduction techniques in the i-vector representation paradigm.

i) Principal component analysis

Principal component analysis (PCA) [106] is a commonly used unsupervised learning technique, which reduces feature vector dimensionality by means of feature decorrelation. A principal component analysis suppresses linear correlations between the interrelated variables by projecting the data onto the direction of the data's maximum variance, so as to minimize the projection distance [107]. Projection distance is defined as the distance between the data points and their projections.

To perform a PCA in the i-vector-based language recognition [105], given K training i-vectors $\boldsymbol{u}_i, i = 1,2,\ldots,K$, a sample covariance matrix of the training data $\boldsymbol{\Sigma}$ is first estimated as follows:

$$\boldsymbol{\Sigma} = \frac{1}{K}\sum_{i=1}^{K}(\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{u}_i - \overline{\boldsymbol{u}})^\top \tag{4.5}$$

where $\overline{\boldsymbol{u}}$ is the training set mean,

$$\overline{\boldsymbol{u}} = \frac{1}{K}\sum_{j=1}^{K}\boldsymbol{u}_j. \tag{4.6}$$

The eigenvalues and eigenvectors can then be obtained by decomposing the sample covariance matrix in (4.5) as follows [107]:

$$\boldsymbol{\Sigma}\boldsymbol{O}_i = \lambda_i \boldsymbol{O}_i, \quad i = 1,\ldots,d, \tag{4.7}$$

where $\lambda_i \geq 0$ are the eigenvalues and the columns of the matrix $\boldsymbol{O}$ correspond to the eigenvectors $\boldsymbol{o}_i$.

Next, the eigenvalues are arranged in decreasing order, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_d$ are found. The eigenvector $\boldsymbol{o}_1$ corresponding to the largest eigenvalue provides the direction of projection where the variance of the data is maximized. Accordingly, $\boldsymbol{o}_2$ is the second direction where the variance is maximized and which is also orthogonal (perpendicular) to $\boldsymbol{o}_1$. Similarly, all $\boldsymbol{o}_i$'s are mutually orthogonal.

In order to perform a dimensionality reduction, the first $\widehat{d}$ ($\widehat{d} < d$) eigenvectors must be selected. They form a $\widehat{d}$-dimensional subspace represented by a matrix $\boldsymbol{B}_{\text{PCA}}$ of size $(d \times \widehat{d})$, in which each column is a principal component. Then, a given i-vector $\boldsymbol{u} \in \mathbb{R}^R$ is projected to this subspace according to [107]:

$$\boldsymbol{u}_{\text{PCA}} = \boldsymbol{B}_{\text{PCA}}^{\top} \boldsymbol{u}, \tag{4.8}$$

where $\boldsymbol{u}_{\text{PCA}}$ is the $\widehat{d}$-dimensional channel compensated i-vector.

ii) Linear discriminant analysis

Linear discriminant analysis (LDA) is a general-purpose technique for dimensionality reduction in various pattern recognition tasks. An LDA was also used in [51] as a channel compensation technique. It projects the data onto a maximum $(M-1)$-dimensional subspace, where M denotes the number of target classes. An LDA aims at maximizing the between-class variation while minimizing the intra-class variation. Specifically, an LDA is meant to optimize the Fisher's objective function defined as [101]:

$$J(\boldsymbol{v}) = \frac{\boldsymbol{v}^{\top} \boldsymbol{\Sigma}_b \boldsymbol{v}}{\boldsymbol{v}^{\top} \boldsymbol{\Sigma}_w \boldsymbol{v}}, \tag{4.9}$$

where the matrices $\boldsymbol{\Sigma}_b$ and $\boldsymbol{\Sigma}_w$ denote the between- and within-class scatter matrices, respectively. For use in i-vector-based language recognition, the matrices are computed as follows [39]:

$$\boldsymbol{\Sigma}_b = \sum_{i=1}^{M} (\overline{\boldsymbol{u}}_i - \overline{\boldsymbol{u}})(\overline{\boldsymbol{u}}_i - \overline{\boldsymbol{u}})^{\top} \tag{4.10}$$

$$\boldsymbol{\Sigma}_w = \sum_{i=1}^{M} \frac{1}{k_i} \sum_{j=1}^{n_i} (\boldsymbol{u}_j^i - \overline{\boldsymbol{u}}_i)(\boldsymbol{u}_j^i - \overline{\boldsymbol{u}}_i)^{\top}, \tag{4.11}$$

where M is the total number of target languages. The mean of training i-vectors across all languages, $\overline{\boldsymbol{u}}$, is defined as:

$$\overline{\boldsymbol{u}} = \frac{1}{K}\sum_{j=1}^{K}\boldsymbol{u}_j, \tag{4.12}$$

where K is the total number of training i-vectors. The mean of the i-vectors for each language, $\overline{\boldsymbol{u}}_i$, is defined as:

$$\overline{\boldsymbol{u}}_i = \frac{1}{k_i}\sum_{j=1}^{k_i}\boldsymbol{u}_j^i, \tag{4.13}$$

where k_i is the number of training i-vectors in language i. An LDA then seeks a $\boldsymbol{v}$ that maximizes the Fisher's criterion (4.9) [101]:

$$\arg\max_{\boldsymbol{v}} J(\boldsymbol{v}). \tag{4.14}$$

One can show that maximizing the Fisher's criterion is equivalent to solving the following eigenvalue equation [101]:

$$\boldsymbol{\Sigma}_b\boldsymbol{v} = \lambda\boldsymbol{\Sigma}_w\boldsymbol{v}. \tag{4.15}$$

The LDA projection matrix, $\boldsymbol{B}_{\text{LDA}}$, is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of $\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Sigma}_b$ [101]. Since $\boldsymbol{\Sigma}_b$ is the sum of M matrices of rank ≤ 1, $\boldsymbol{\Sigma}_b$ will have a maximum rank of $(M-1)$, indicating that only $(M-1)$ of the eigenvalues λ will be non-zero [101]. Finally, each i-vector $\boldsymbol{u} \in \mathbb{R}^d$ is projected to the new subspace according to [51]:

$$\boldsymbol{u}_{\text{LDA}} = \boldsymbol{B}_{\text{LDA}}^{\top}\boldsymbol{u}, \tag{4.16}$$

where $\boldsymbol{u}_{\text{LDA}}$ represents the LDA channel compensated i-vector with the maximum $(M-1)$ dimension.

iii) Heteroscedastic linear discriminant analysis

An LDA projects the data onto a subspace whose maximum dimensionality is $(M-1)$, where M is the number of classes. These $(M-1)$-dimensional subspaces may not necessarily contain all of

the relevant information needed for a class separation [108]. *Heteroscedastic linear discriminant analysis* (HLDA) [109], an extension of the LDA, is currently used within speech processing for dimensionality reduction and feature representation [110–113]. Unlike the LDA, an HLDA does not assume shared covariance matrix for all classes. Instead, it takes into account the discriminatory information presented in both class means and variances [109, 110].

In the HLDA technique, the feature vectors of dimensionality d are projected into first $q < d$ rows, $\boldsymbol{b}_{j=1\ldots q}$, of the $d \times d$ HLDA transformation matrix, $\boldsymbol{B}_{\text{HLDA}}$. The matrix $\boldsymbol{B}_{\text{HLDA}}$ is estimated by an efficient row-by-row iteration method, whereby each row is iteratively updated as [110, 114]:

$$\hat{\boldsymbol{b}}_j = \hat{\boldsymbol{s}}_j \boldsymbol{G}^{(j)-1} \sqrt{\frac{K}{\hat{\boldsymbol{s}}_j \boldsymbol{G}^{(j)-1} \hat{\boldsymbol{s}}_j^\top}} \,. \tag{4.17}$$

Here, $\hat{\boldsymbol{s}}_j$ is the j^{th} row vector of the *co-factor matrix* $\hat{\boldsymbol{S}} = |\boldsymbol{B}_{\text{HLDA}}| \boldsymbol{B}_{\text{HLDA}}^{-1}$ for the current estimate of $\boldsymbol{B}_{\text{HLDA}}$ and

$$\boldsymbol{G}^{(j)} = \begin{cases} \sum_{i=1}^{M} \frac{k_i}{\boldsymbol{b}_j \boldsymbol{\Sigma}_i \boldsymbol{b}_j^\top} \boldsymbol{\Sigma}_i & j \leq q \\ \frac{K}{\boldsymbol{b}_j \boldsymbol{\Sigma} \boldsymbol{b}_j^\top} \boldsymbol{\Sigma} & j > q, \end{cases} \tag{4.18}$$

where $\boldsymbol{\Sigma}$ denotes the class-independent covariance matrix computed from (4.5), $\boldsymbol{\Sigma}_i$ is the covariance matrix of the i^{th} model, k_i is the number of training i-vectors of the i^{th} class and $K = \sum_{i=1}^{M} k_i$ is the total number of i-vectors. Finally, given i-vector $\boldsymbol{u} \in \mathbb{R}^d$, the HLDA channel compensated i-vector $\boldsymbol{u}_{\text{HLDA}}$ is obtained by [110]:

$$\boldsymbol{u}_{\text{HLDA}} = \boldsymbol{B}_{\text{HLDA}}^\top \boldsymbol{u}. \tag{4.19}$$

Within-class covariance normalization

Along with an i-vector dimensionality reduction, a *within-class covariance normalization* (WCCN) [100] can be applied for the compensation of unwanted intra-class variations within the total variability space [51]. The WCCN projection matrix, $\boldsymbol{B}_{\text{WCCN}}$, is obtained by a Cholesky decomposition of $\boldsymbol{B}_{\text{WCCN}} \boldsymbol{B}_{\text{WCCN}}^\top = \boldsymbol{\xi}^{-1}$ from

the dimensionality reduced i-vectors, where a within-class covariance matrix, $\boldsymbol{\xi}$, is computed according to [100]:

$$\boldsymbol{\xi} = \frac{1}{M}\boldsymbol{\Sigma}_w, \tag{4.20}$$

where M is the total number of target languages and $\boldsymbol{\Sigma}_w$ was defined earlier in (4.11). Given i-vector $\boldsymbol{u}$, the WCCN channel compensated i-vector $\boldsymbol{u}_{\text{WCCN}}$ is obtained by [51]:

$$\boldsymbol{u}_{\text{WCCN}} = \boldsymbol{B}_{\text{WCCN}}^{\top}\boldsymbol{u}. \tag{4.21}$$

4.3 SCORING AND NORMALIZATION

Classification techniques, such as SVMs [115], Gaussian scoring [115], PLDA scoring [116] and cosine similarity scoring [39] are typically used for i-vector-based language recognition tasks. In publications [**I–IV**], cosine similarity scoring was adopted to perform classification.

Following an i-vector dimensionality reduction using an HLDA, a WCCN is further applied in [**IV**]. Denoting the HLDA and WCCN projection matrices by $\boldsymbol{B}_{\text{HLDA}}$ and $\boldsymbol{B}_{\text{WCCN}}$, respectively, the compensated i-vectors are obtained by:

$$\hat{\boldsymbol{u}} = \boldsymbol{B}_{\text{WCCN}}^{\top}\boldsymbol{B}_{\text{HLDA}}^{\top}\boldsymbol{u}. \tag{4.22}$$

Then, the cosine similarity scoring between two compensated i-vectors $\hat{\boldsymbol{u}}_{\text{target}}$ and $\hat{\boldsymbol{u}}_{\text{test}}$ is defined as follows [39]:

$$\text{score}(\hat{\boldsymbol{u}}_{\text{target}}, \hat{\boldsymbol{u}}_{test}) = \frac{\hat{\boldsymbol{u}}_{\text{target}}^{\top}\ \hat{\boldsymbol{u}}_{\text{test}}}{\|\hat{\boldsymbol{u}}_{\text{target}}\|\ \|\hat{\boldsymbol{u}}_{\text{test}}\|}, \tag{4.23}$$

where $\hat{\boldsymbol{u}}_{\text{target}}$ is the average i-vector over all of the training utterances of the target class i,

$$\hat{\boldsymbol{u}}_{\text{target}} = \frac{1}{k_i}\sum_{i=1}^{k_i}\hat{\boldsymbol{u}}_i, \tag{4.24}$$

and $\hat{\boldsymbol{u}}_{test}$ is a test i-vector and k_i is the number of training i-vectors in class i.

After obtaining scores $\{t_i, i = 1, \ldots, M\}$ for a particular test utterance of class i, scores are further post-processed as [4]:

$$\hat{t}_i = \log \frac{\exp(t_i)}{\frac{1}{M-1}\sum_{j \neq i} \exp(t_j)}, \tag{4.25}$$

where $\hat{t}_i$ is the detection log-likelihood ratio for a particular test utterance of class i, scored against all of the M target classes.

5 Attribute-based foreign accent recognition

In Chapter 2, we described the cues which are beneficial in discriminating between different foreign accents. In this chapter, we focus on another type of linguistic variation which occurs when non-native speakers use the characteristics of their L1 in their L2, specifically, a variation in *speech attributes* [117]. When humans listen to a particular language without having a linguistic knowledge of the language, they learn to identify *fundamental* speech cues in that language. For example, one can identify the tonal nature of Mandarin Chinese or Vietnamese. Speech attributes, also known as *articulatory features* [118], are a set of *universal language descriptors* shared across languages and assumed to be less language-dependent than phones [119, 120].

Speech attributes can be divided into three broad categories: *manner of articulation*, *place of articulation* and *voicing* [118]. The manner of articulation describes the interaction between the speech organs such as the jaw, tongue and lips, in making a sound [118]. The following manners of articulation are used in the experiments of this dissertation:

- **Fricative**: A fricative consonant is articulated by bringing the mouth into a position, where the air passes through a small gap. Examples of English fricative sounds are /f/ and /v/ as in *from* and *have*, respectively.
- **Glide**: Glides, also known as semi-vowels, are consonants which have vowel-like articulation, but with a narrower constriction in the vocal tract. Examples of English glides are /j/ and /w/ as in *yes* and *wish*, respectively.
- **Nasal**: Nasal consonants are produced when the air flow is

blocked through the mouth and the air passes through the nose. Examples of English nasal sounds are /m/ and /n/ as in *mom* and *no*, respectively.

- **Stop**: Stop sounds are those consonants which at some point during the articulation, the air flow is blocked and then released. Examples of English stop sounds are /b/ and /p/ as in *bee* and *pose*, respectively.
- **Vowel**: A vowel is a sound produced when articulators hold a shape with no constriction in the vocal tract. Examples of English vowels are /a:/ and /i:/ as in *father* and *fleece*, respectively.

In contrast, the place of articulation indicates the position, where obstructions in the vocal tract occur [118]. Figures 5.1 and 5.2 illustrate the position of common places of articulation in the human vocal tract. The following places of articulation are used in the experiments of this dissertation:

- **Coronal**: It refers to the consonants which use the front part of the tongue for articulation, as illustrated in Figure 5.1. Examples of English coronal sounds are /s/ and /t/ as in *seal* and *team*, respectively.
- **Dental**: Dental consonants are produced when the air flow is constricted by placing the tongue against the upper teeth, as illustrated in Figure 5.1. Examples of English dental sounds are /θ/ and /ð/ as in *bath* and *the*, respectively.
- **Glottal**: Glottal sounds happen at the glottis, that is the part of larynx consisting of the vocal cords, by bringing the vocal cords together, as illustrated in Figure 5.1. /h/ as in *hi*, is an example of English glottal sound.
- **Labial**: In labial sounds, one or both lips are actively engaging in articulation, as illustrated in Figure 5.1. Examples of labial consonants are /v/ and /b/ as in *voice* and *beef*, respectively.

- **Low**: It refers to the sounds with a relatively wide space between the tongue and the roof of the mouth (palate), as illustrated in Figure 5.2. Examples of low English vowels are /a:/ and /æ/ as in *arm* and *sat*, respectively.

- **High**: It refers to the sounds with a relatively narrow space between the tongue and the roof of the mouth, as illustrated in Figure 5.2. /i:/ as in *see*, is an example of high English vowel.

- **Mid**: It refers to the sounds which space between the tongue and the roof of the mouth is approximately between low and high, as illustrated in Figure 5.2. /e/ as in *pet*, is an example of mid English vowel.

- **Retroflex**: In retroflex consonants, the tongue articulates with the hard palate, as illustrated in Figure 5.1. English /r/ as in *road*, is an example of Retroflex sound.

- **Velar**: In velar sounds, the back of the tongue articulates with the soft palate, as illustrated in Figure 5.1. Examples of English velar consonants are /k/ and /g/ as in *book* and *good*, respectively.

Finally, voicing refers to either vibration or non-vibration of the vocal cords. For example, English consonants /s/ and /t/ are both voiceless sounds, while /s/ is a fricative sound and /t/ is a stop sound [118].

Speech attributes were used in *automatic speech attribute transcription* (ASAT) framework [122] in an attempt to mimic human speech recognition (HSR) capabilities and consequently bridge the gap between the performance of ASR and HSR [123]. The primary aim of ASAT was to provide additional information to the ASR by integrating acoustic and phonetic knowledge in the form of speech attributes using data-driven modeling techniques. Later, articulatory features were adopted into the acoustic modeling of context-independent phone models in [124]. A universal phone recognizer

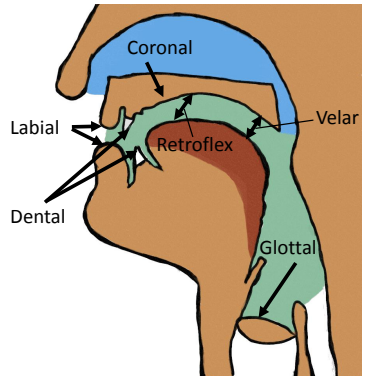


Figure 5.1: Position of common places of articulation in the human vocal tract. Re-drawn by the author of the thesis, with inspiration taken from [118].

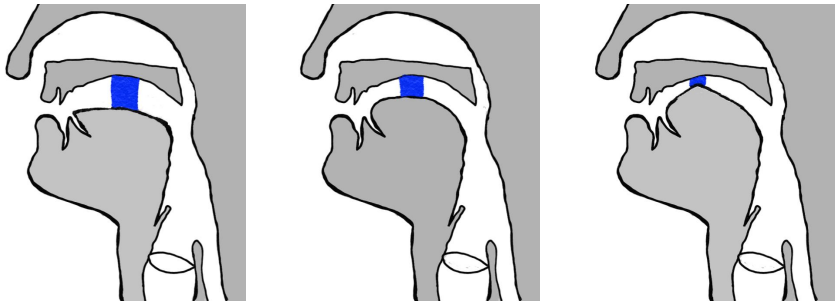


Figure 5.2: Relative height distinctions among English sounds: (a) Low, (b) Mid, and (c) High. Re-drawn by the author of the thesis, with inspiration taken from [121].

using a bank of attribute detectors was proposed in [125] for processing the speech signal of all languages including never-seen languages. The bank of speech detectors was trained by sharing language specific data with no loss in phone recognition performance. In [126], speech attributes, in the form of manner and place of articulation, were chosen to characterize spoken languages similar to in the ASAT framework. Spoken utterances were first tokenized

into sequences of universal speech attributes. Then, feature vectors were generated using the statistics of co-occurrences of manner and place units by considering the tokenized spoken utterance as a spoken document. Finally, vector space language classifier [126] was adopted in order to make language recognition decisions.

In this dissertation, the author adopts a universal attribute characterization of speech signals for the foreign accent recognition task. Before discussing the details of the proposed system, a few examples are first provided to indicate the usefulness of speech attributes in characterizing foreign spoken accents. These examples are selected from the languages used in experiments of this thesis. For example, /p/ is not found at the beginnings of the words in Vietnamese [127]. Vietnamese speakers of English may substitute a fricative sound, such as /b/ or /f/, when producing the English stop sound /p/ at the beginning of words [127]. For example, the word *put* may be pronounced as /fʊt/ [127]. In this case, detecting a fricative in the beginning of the word is a potential foreign accent cue for Vietnamese L1. Further, Vietnamese speakers may omit fricatives such as /f/, /v/ and /s/ at the end of words when speaking English, since fricatives do not occur in a word's final position in their mother tongue [128]. For example, *beef* may be pronounced as /biː/ or the sentence *The boys always pass the garage on their way home* may sound like *The boy alway pa the gara on their way home* [129].

Similarly, /ŋ/, which is a nasal sound in English, does not exist in Russian [130]. This makes it difficult for Russian speakers to correctly pronounce the sentence *The singer sang a nice song*. Cantonese consonants, except nasals and semi-vowels, are all voiceless, making it challenging for Cantonese speakers to pronounce English voiced consonants [131], resulting in a foreign spoken accent.

Speech attributes provide a unified approach by combining both phonotactically-inspired and spectral approaches for the problem of foreign accent recognition [132, 133]. Lack of transcribed accent-specific speech data may hinder the use of phonotactic-based techniques in foreign accent recognition tasks [134, 135]. As discussed in the previous chapters, phonotactic features provide useful cues

for discriminating between languages. Thus, having a unified approach that takes advantage of both acoustic and phonotactic information is expected to be beneficial for the task of foreign accent recognition.

Figure 5.3 displays the block diagram of the attribute based foreign accent recognition system proposed in this thesis. In this system, short-term spectral features are fed into left- and right-context ANNs and merged to form speech attribute posterior probabilities. Then, a PCA is adopted to capture long-term contextual information from the consecutive speech frames of the posterior probabilities. Obtained feature streams are then modeled with the i-vector approach [51] followed by cosine scoring [136].

5.1 SPEECH ATTRIBUTE EXTRACTION

The process of speech attribute extraction consists of converting an input speech utterance into feature streams, where each element corresponds to the level of presence or level of activity of a particular property of an attribute over time [123]. A bank of speech detectors, each designed for detecting a particular speech attribute, is used for the experiments in this thesis (Figure 5.3). Each detector consists of three single hidden layer feed-forward ANNs with a hierarchical structure. A window of 310 ms, centered around each speech frame, is split into two halves to form left- and right-context speech frames. Sub-band energy trajectories are extracted from each half with a 15-band mel-frequency filterbank and fed into the corresponding left- and right-context ANNs. The third ANN merges the outputs of these two independent ANNs and results in the posterior probabilities of the target speech attribute.

Similar to the conventional HMM approach to the ASR, each attribute is modeled as having multiple states [123]. It was shown in [124] that better phone recognition results are obtained by considering three states per each attribute. Similar to [137], each attribute contains three states, namely *attribute present* (target), *attribute absent* (non-target) and *noise* [124] in experiments of this thesis. Given a

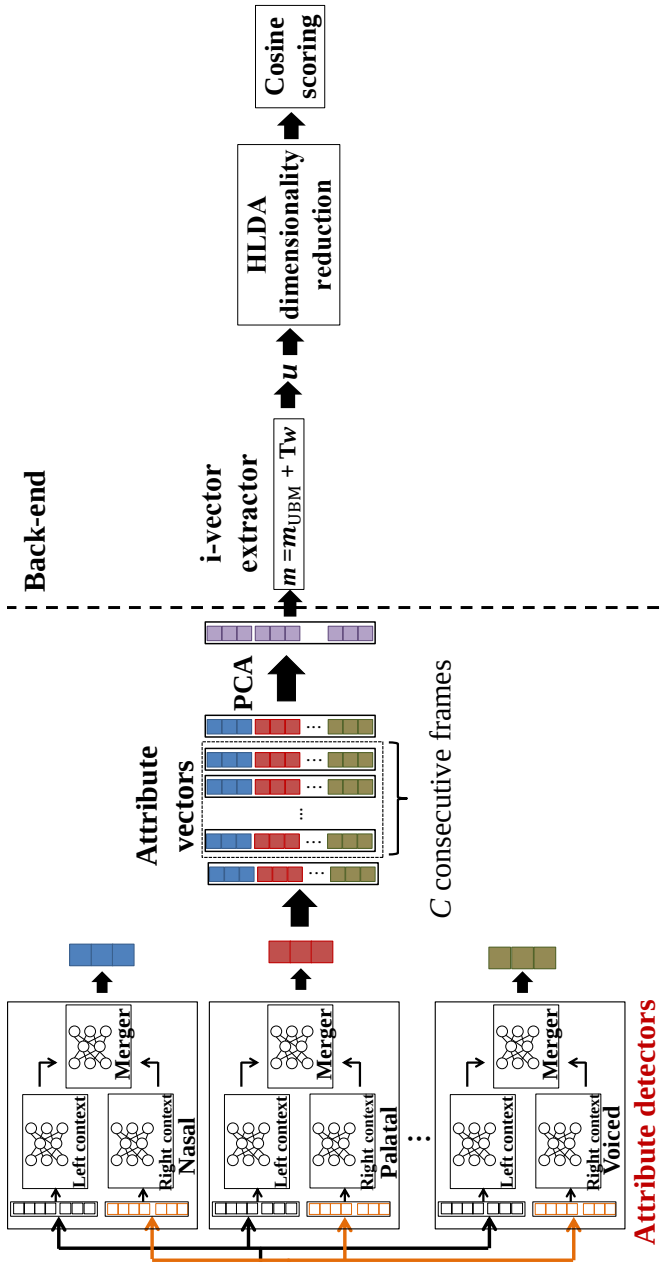


Figure 5.3: Block diagram of the proposed attribute-based foreign accent recognition system (from [IV]). Spectral features are fed into left- and right-context artificial neural networks and merged by a third neural network to form the final attribute posterior probabilities. A PCA is then applied on the consecutive frames of these posterior probabilities in order to capture long-term contextual information. The back-end consists of i-vector approach followed by a foreign accent classifier. In this Figure, $\overline{L}$ denotes the speech frame length, C is the PCA context size, and t and f represent time and frequency, respectively.

speech frame $\boldsymbol{f}$ and a target attribute class i, each attribute detector outputs three posterior probabilities (speech events), $p(H^{(i)}_{\text{target}}|\boldsymbol{f})$, non-target (denoted as anti), $p(H^{(i)}_{\text{anti}}|\boldsymbol{f})$, and noise, $p(H^{(i)}_{\text{noise}}|\boldsymbol{f})$. These probabilities add up to one for each frame. The final feature stream then, is formed by stacking the posterior probabilities delivered by each attribute detector into a supervector of attribute detection scores, as indicated in Figure 5.3.

The set of speech attributes used in the experiments of this thesis are five manner of articulation classes (fricative, glide, nasal, stop and vowel) and nine place of articulation classes (coronal, dental, glottal, high, labial, low, mid, retroflex and velar) together with a voicing class. The final feature vector for manner and voice comprises 18 dimensions (6 attributes $\times$ 3) and for place, it comprises 27 dimensions (9 attributes $\times$ 3) for each speech frame.

5.2 TRAINING ATTRIBUTE DETECTORS

Each attribute detector is realized with three independent feed-forward multi-layer perceptrons (MLPs) each having one hidden-layer and 500 hidden nodes with a sigmoidal activation function. In this thesis, we used the "stories" part of the OGI multi-language telephone speech corpus [138] to train the attribute detectors. This corpus consists of phoneme transcriptions for six languages: English, German, Hindi, Japanese, Mandarin, and Spanish. For our purposes, data from each language was pooled together totaling approximately 5 hours and 34 minutes of training and 31 minutes of validation data. The training data is categorized into "attribute present", "attribute absent" and "other" regions for every attribute class. Manner and place attributes were annotated using the phoneme transcriptions provided in the OGI corpus. Annotation examples are displayed in Table 5.1 for the English portion. For training the MLP parameters, back-propagation algorithm with a cross-entropy cost function [77] is used. The reduction in classification errors on a development set is used as the stopping criterion in order to prevent over-fitting [137].

Table 5.1: Examples of manner and place annotations for English phonemes in the OGI corpus adopted from [137]. Transcriptions are based on TIMIT [139] phoneme transcription.

	Attributes	Phonemes
Manner of articulation	Vowel	/iy/, /ih/, /eh/, /ey/, /ae/, /aa/, /aw/, /ay/, /ah/, /oy/, /ow/, /uh/, /uw/, /er/
	Fricative	/jh/, /ch/, /s/, /sh/, /z/, /f/, /th/, /v/, /dh/, /hh/
	Nasal	/m/, /n/, /ng/
	Stop	/b/, /d/, /g/, /p/, /t/, /k/, /dx/
	Approximant	/w/, /y/, /l/, /r/, /er/
Place of articulation	Coronal	/d/, /dx/, /l/, /n/, /s/, /t/, /z/
	High	/ch/, /ih/, /iy/, /jh/, /sh/, /uh/, /uw/, /y/, /ey/, /ow/, /g/, /k/, /ng/
	Dental	/dh/, /th/
	Glottal	/hh/
	Labial	/b/, /f/, /m/, /p/, /v/, /w/
	Low	/aa/, /ae/, /aw/, /ay/, /oy/, /ah/, /eh/
	Mid	/ah/, /eh/, /ey/, /ow/
	Retroflex	/er/, /r/
	Velar	/g/, /k/, ng/

5.3 LONG-TERM SPEECH ATTRIBUTE EXTRACTION

Since language and accent recognition systems generally benefit from including long-term contextual information [140, 141] in acoustic feature vectors, we explore the same idea when considering speech attribute modeling. One way of doing this is to treat the attributes analogous to cepstral coefficients and compute the SDC-like features. In [**II**], contextual information in forms of delta and double delta features were computed from attribute features. We also introduced another approach for extracting contextual information from attribute feature streams with the help of a PCA in [**II**,**IV**]. Specifically, let $\boldsymbol{x}(t)$ represent 18-dimensional (6 manner attributes $\times$ 3) or 27-dimensional (9 place attributes $\times$ 3) speech attribute feature vectors at frame t. Given a context of size C, a sequence of $d = 18 \times C$ (or $d = 27 \times C$, for place) dimensional stacked vectors $\tilde{\boldsymbol{x}}_C(t) = (\boldsymbol{x}(t)^\top, \boldsymbol{x}(t+1)^\top, \ldots, \boldsymbol{x}(t+C-1)^\top)^\top, t = 1, 2, \ldots,$

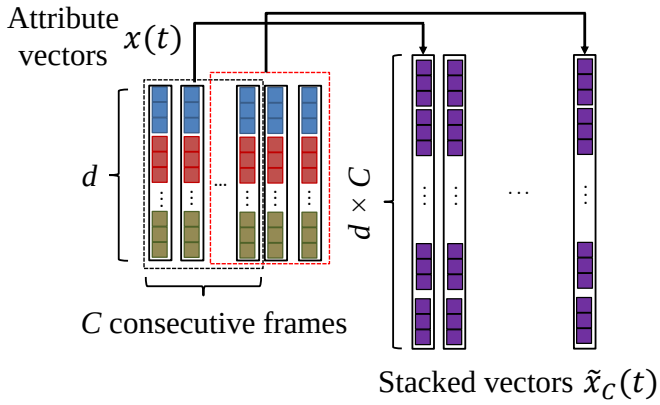


Figure 5.4: Stacking d-dimensional speech attributes at frame t using a context of size C. Given C, a sequence of $d = 18 \times C$ (or $d = 27 \times C$, for place) dimensional stacked vectors $\tilde{\boldsymbol{x}}_C(t) = (\boldsymbol{x}(t)^\top, \boldsymbol{x}(t+1)^\top, \ldots, \boldsymbol{x}(t+C-1)^\top)^\top, t = 1, 2, \ldots$, is formed in order to capture long-term contextual information.

is formed, as illustrated in Figure 5.4. Each $\tilde{\boldsymbol{x}}_C(t)$ is projected to the largest eigenvalues of the sample covariance matrix using PCA to preserve 99 % of the cumulative variance. A PCA projection matrix is estimated from the same data as the UBM in our experiments. The final dimensionality, after a PCA projection, varies from approximately 20 to 100, where longer contexts have larger dimensionality.

6 Performance evaluation

An important aspect of any speech processing task is to evaluate performance using standard protocols. During the past several years, NIST has provided the speech research community with standard evaluation metrics and datasets for several tasks, including language recognition evaluations in 1996, 2003, 2005, 2007, 2009, 2011 and 2015. The focus of the NIST LREs has been on detection tasks, both in closed-set and open-set identification problems. In the detection tasks defined by the NIST, given a speech segment and a language hypothesis (i.e. target language of interest), the system decides whether a hypothesized language is spoken in a given test segment [142]. In open-set identification problems, the language of a test segment might not be any of the indicated target languages [4].

Here, we describe the evaluation metrics, i.e. detection cost and detection error tradeoff (DET) curve, adopted within our foreign accent recognition task in **[I–IV]**. The performance measure of an open-set LID task used in **[V]** is also presented.

Average detection cost

The *average detection cost* is defined as [142]:

$$C_{\text{avg}} = \frac{1}{M} \sum_{i=1}^{M} C_{\text{DET}}(y_i), \tag{6.1}$$

where $C_{\text{DET}}(y_i)$ is the detection cost for a subset of test segment trials for which the target language is y_i and M is the number of target languages. The cost per each target language is defined as [142]:

$$\begin{aligned} C_{\text{DET}}(y_i) &= C_{\text{miss}} p_{\text{tar}} p_{\text{miss}}(y_i) \\ &+ C_{\text{fa}}(1 - p_{\text{tar}}) \frac{1}{M-1} \sum_{k \neq j} p_{\text{fa}}(y_j, y_k), \end{aligned} \tag{6.2}$$

where the *miss* probability (or *false rejection rate*) is denoted by p_{miss}, representing the error of rejecting a test segment in y_i that was actually spoken in that language. Similarly, $p_{\text{fa}}(y_j, y_k)$ denotes the *false alarm* (or *false acceptance*) probability when a test segment in language y_k is misclassified as being in language y_j. This is computed for each target and non-target language pair. The probabilities are calculated by dividing the number of errors by the total number of trials in each subset. Combining (6.1) and (6.2), the average detection cost can be represented as [142]:

$$\begin{aligned} C_{\text{avg}}(\theta_{\text{threshold}}) &= C_{\text{miss}} p_{\text{tar}} \underbrace{\frac{1}{M}\sum_{i=1}^{M} p_{\text{miss}}(y_i)}_{p_{\text{miss}}(\theta_{\text{threshold}})} \\ &+ C_{\text{fa}}(1-p_{\text{tar}}) \underbrace{\frac{1}{M}\sum_{i=1}^{M}\left[\frac{1}{M-1}\sum_{k\neq j} p_{\text{fa}}(y_j, y_k)\right]}_{p_{\text{fa}}(\theta_{\text{threshold}})}, \end{aligned} \tag{6.3}$$

where $\theta_{\text{threshold}}$ is a threshold for making hard decisions in order to compute the detection cost in (6.3). In this threshold-based decision, the assumption is that the higher detection scores favor the target language hypothesis, while the lower scores favor the alternative. For a general decision threshold with $\theta_{\text{threshold}}$ being fixed across all of the language pairs, the minimum of C_{avg} over all $\theta_{\text{threshold}}$ is defined as min C_{avg} [142].

In all of the experiments of this thesis, the costs, C_{miss} and C_{fa} are both set to 1 and p_{tar}, the prior probability of a target language, is set to 0.5 following [142]. For computing detection costs, FoCal multi-class toolkit was used [143].

Detection error tradeoff (DET) curve

As previously discussed, computing the detection cost in (6.3) is performed by setting a threshold for the detection scores. The threshold can vary across a range of possible operating points. One can plot the $p_{\text{miss}}(\theta_{\text{threshold}})$ against $p_{\text{fa}}(\theta_{\text{threshold}})$ for different values of $\theta_{\text{threshold}}$ using normal deviate scale. The resulting graph represents the *detection error tradeoff* (DET) curve [144]. An example of a

DET plot is displayed in Figure 6.1. The curve becomes a straight line when target and non-target scores are normally distributed [4]. The error at the operating point, at which the false alarm and the miss alarm probabilities are equal, is known as the EER.

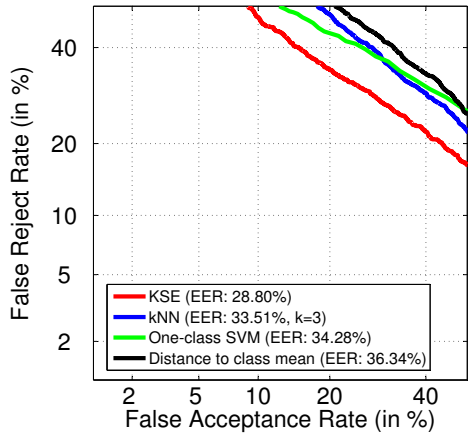


Figure 6.1: An example of a DET curve adopted from [**V**] comparing the performance of the proposed OOS detection (KSE) and three different baseline methods. Equal error rate (EER) is the point on the DET curve at which the false acceptance rate and false rejection rate are equal.

Performance measure in an open-set LID

In [**V**], we followed the performance measure of an open-set LID task defined in the NIST 2015 language recognition i-vector challenge, computed as follows [88]:

$$\text{Cost} = \frac{(1 - p_{\text{OOS}})}{M} \sum_{i=1}^{M} p_{\text{miss}}(y_i) + p_{\text{OOS}} \times p_{\text{miss}}(\text{OOS}), \qquad (6.4)$$

where $p_{\text{miss}}(y_i)$ and $p_{\text{miss}}(\text{OOS})$ are the error rates of the test segments not assigned to the correct language y_i or OOS class, respectively. In the i-vector challenge [88], constant p_{OOS} was set to 0.230 and the number of target languages was $M = 50$. Substituting these

values into (6.4) provides the following:

$$\text{Cost} = \sum_{i=1}^{50} 0.0154 \times p_{\text{miss}}(y_i) + 0.230 \times p_{\text{miss}}(\text{OOS}). \tag{6.5}$$

This indicates that the cost of misclassifying an OOS segment as an inset is much higher than the opposite, in this task.

7 Summary of publications and results

This chapter summarizes the contributions and significant results of the five, previously mentioned publications. Publications [**I**], [**II**] and [**IV**] address the problem of foreign accent recognition. Publication [**III**] investigates the problem of dialect leveling within the context of Finnish dialect identification. Publication [**V**] addresses OOS data selection within the context of an open-set LID.

Figure 7.1 depicts how each publication, except [**III**] that focuses on dialect recognition, contributes to the overall proposed foreign accent recognition system. Publications [**II**,**IV**] contribute to the front-end processing of the spoken foreign accents, while [**I**] focuses on the back-end and [**V**] on the open-set identification cases. It should be noted that although [**V**] investigates the OOS data selection in an open-set LID problem, it can be also adopted in an open-set foreign accent recognition.

Table 7.1 summarizes the corpora used in the publications. The FSD corpus [145] was originally developed to assess Finnish language proficiency among adults of different nationalities. From this corpus, we selected 8 accents – Russian, Albanian, Arabic, English, Estonian, Kurdish, Spanish, and Turkish – with a sufficient number of utterances available (accents with more than 70 utterances). From the NIST 2008 SRE corpus [146], 7 accents – Hindi, Thai, Japanese, Russian, Vietnamese, Korean and Chinese Cantonese – with enough available data were selected. These accents are from the short2, short3 and 10 second portions of the original corpus. Finally, the speech data in the SAPU (*Satakunta in Speech*) corpus [147] were recorded in Satakunta, in South-Western Finland between 2007–2013 in an interview setting with topics relating to informants' lives and home regions.

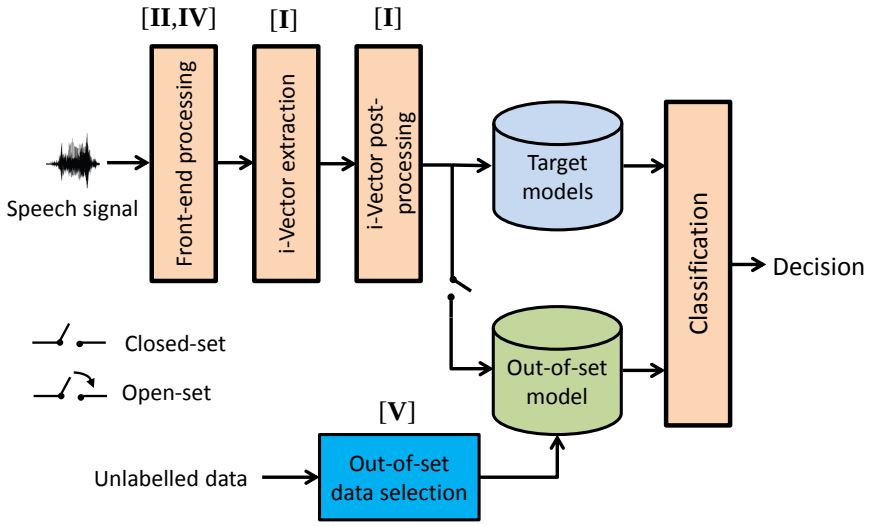


Figure 7.1: The overall foreign accent recognition system with an indication of how each publication contributes to the system.

7.1 SUMMARY OF PUBLICATIONS

In **[I]**, we have performed an extensive experiment with an i-vector based foreign accent recognition system in order to assess the effects of three different aspects of the recognition system: (1) recognition system parameters, (2) data used for estimating the system hyper-parameters and (3) language aspects, including a study of confusion patterns among different accents and comparing the individual accents in terms of their detection difficulty. First, we have studied the various choices of i-vector extraction hyper-parameters including the UBM size, the i-vector dimensionality and the effectiveness of an HLDA dimensionality reduction algorithm. Furthermore, we have studied how the choices of dataset in training the i-vector hyper-parameters, namely, the UBM, the T-matrix and the HLDA projection matrix, affect the foreign accent recognition accuracy. If this effect had been trivial in regards to overall recognition accuracy, then it would have been easy to skip the computationally expensive steps of the UBM and especially, the i-vector extractor

Table 7.1: A summary of the corpora used in the publications.

Corpus	Summary	Usage
Finnish (FSD) [145]	A total of 1,644 interview utterances corresponding to 415 speakers and 8 foreign accents are collected. Raw audio files are partitioned into 30 second portions and resampled to 8 kHz.	Foreign accent recognition in [**I**,**II**,**IV**]
English (NIST 2008 SRE) [146]	A total of 1,262 telephone recorded utterances corresponding to 315 speakers and 7 foreign accents are collected. The sampling rate is 8 kHz.	Foreign accent recognition in [**IV**]
Finnish regional dialects (SAPU) [147]	The corpus consists of 282 utterances (231 hours 31 minutes) and 8 dialects recorded in interview settings. Raw audio files are partitioned into 30 second portions and resampled to 8 kHz.	Dialect Leveling in [**III**]
OGI Multi-language telephone speech [138]	This corpus contains phonetic transcriptions of six languages. Data from each language is pooled together to obtain approximately 5 hours and 34 minutes of training and 31 minutes of validation data, in total.	Training the attribute detectors in [**II**, **III**, **IV**]
NIST language i-vector challenge [88]	The corpus consists of 50 target languages corresponding to 15,000 training utterances. Test and development sets contain 6,500 unlabeled i-vectors each. The i-vectors are derived from conversational telephone and narrow-band broadcast speech data.	Open-set language identification in [**V**]

training in the i-vector extraction process. Finally, in our last experiment, we have related the foreign accent recognition accuracy to four affecting factors, namely, Finnish language proficiency, age of entry, level of education and where the second language is spoken.

In [**II**], we have proposed a hybrid acoustic and phonotactically-inspired approach to the problem of foreign accent recognition. This approach was based on defining a common set of fundamental units, known as speech attributes, which are shared across different spoken languages. These units were adopted to represent the foreign accent cues in each spoken utterance. Although these units exist in all languages, the statistics of their co-occurrences vary from one language to another, motivating their use in foreign accent recognition. In this study, shallow neural networks have been adopted to extract the manner attributes from each speech utterance. Furthermore, contextual information in the forms of delta and double delta features computed from the attribute features were appended to the attribute feature vectors. Following this, the extracted attribute features were modeled with the state-of-the-art i-vector methodology.

In [**III**], the author has adopted the attribute-based characterization of speech utterances in order to study dialect leveling — a phenomenon in which the spoken dialects within a language get closer to the standard spoken dialect of that language. Leveling can be considered a nuisance factor in automatic dialect recognition problems where similar spoken dialects are more difficult to discriminate. In this study, three leveling effects were studied, namely, age, gender and region of birth. We have hypothesized that leveling is more pronounced in younger speakers, while older speakers might still preserve properties of their regional dialects. We have addressed the leveling phenomenon on Finnish regional dialects (SAPU corpus) containing speech material collected in the Satakunta region (South-Western Finland) between 2007 and 2013.

In [**IV**], the author has considerably expanded upons the preliminary findings of automatic foreign accent recognition achieved in [**II**] in a more systematic and organized way with the goal of explaining why speech attribute feature extraction followed by an i-vector back-end scoring, is useful in discriminating between foreign accents. The key experiments not included in this earlier study were (1) investigating the effect of HLDA on the foreign accent recogni-

tion accuracy by contrasting it against LDA, (2) exploring the effect of training and test file durations on the foreign accent recognition results, (3) experimenting with an English foreign accented speech corpus together with Finnish data, (4) experimenting with the place of attributes in the results and finally, (5) exploring the effect of feature level fusion on foreign accent recognition results.

In [**V**], the author has proposed a new approach to the problem of OOS data selection using i-vectors within the context of an open-set LID. In this approach, each unlabeled i-vector was given a per-class outlier score with the help of a non-parametric KS test. This score represented the confidence that an i-vector corresponds to OOS data. Out-of-Set candidates were detected from an unlabeled development set and then used to train an additional model to represent the OOS languages in the back-end.

7.2 SUMMARY OF RESULTS

[**I**]: Training hyper-parameters from mismatched dataset results in greatly degraded recognition results in comparison to training them from the application-specific dataset. The most effective hyper-parameter settings that resulted in the highest accent recognition accuracy are UBMs with 512 Gaussians, i-vector dimensionality of 1000 and an HLDA dimensionality of 180. An analysis of the affecting factors suggests that (1) mother tongue traits are relatively more pronounced in older speakers when speaking a second language than in younger speakers. Detection cost decreases relatively by 16% from the age group [11–20] years to [61–70] years and (2) mother tongue detection from speakers with a higher proficiency in Finnish is more difficult than those with a lower proficiency. The highest foreign accent accuracy is attributed to speakers with the lowest grade, while the lowest accuracy is attributed to speakers with the highest grade.

[**II**]: The proposed foreign accent recognition technique based on manner of articulation achieves a 16% relative reduction in EER

over the state-of-the-art SDC-MFCC[1] i-vector baseline system. Concatenating contextual information in the forms of delta and double delta features to manner feature streams results in a further 12% relative reduction in EER over the manner-based system.

[III]: The results indicate that dialect recognition accuracy in the younger age group is considerably lower than in the older age group, suggesting that the dialect in the younger group has leveled. Further, the results suggest that the manner of articulation system outperforms the baseline SDC-MFCC system[2] in the younger age group by a 32% relative decrease in the detection cost. This suggests that an attribute-based system is more robust against age-related leveling effects within the younger group.

[IV]: The findings of this publication can be summarized as follows: (1) Foreign accent recognition results of the FSD corpus suggest that the best performance is obtained by using manner attribute features with i-vector methodology, yielding 45% and 15% relative reductions in the average detection cost over the conventional GMM-UBM and state-of-the-art i-vector approaches with SDC-MFCC features, respectively; (2) Analysis on the effect of training set size and test utterance length reveals that the attribute-based foreign accent recognition system outperforms the spectral-based system in all of the evaluated cases (independently of the amount of training data and test utterance length); (3) A 14% relative reduction in the detection cost is obtained by incorporating contextual information; (4) Manner- and place-based systems outperform the SDC-MFCC-based i-vector system for English data, yielding 15% and 8% relative reductions in the average detection cost, respectively; (5) Concatenating SDC-MFCC features with the attribute feature streams further improves foreign accent recognition results and finally, (6) When comparing the performance between the proposed attribute system and the state-of-the-art spectral i-vector system, the

[1]Acoustic SDC-MFCC features are formed by appending MFCC features to SDC features.

[2]The baseline SDC-MFCC system is realized with the GMM-UBM approach using acoustic SDC-MFCC features.

Table 7.2: Summary of the previous studies on foreign accent recognition. CR = classification rate (in %), EER = equal error rate (in %).

Method (features+modeling)	Language (#accents)	Evaluation
Phonemes+Trajectory [148]	English (5)	90 CR (pair-wise)
Spectral+SVM [149]	Chinese+Spanish (25)	17.50 EER, ∼60 CR
Spectral+i-vector [150]	English (5)	58 CR
Phonetic+GMM-UBM [113,151]	English (7)	∼54 CR
Spectral+i-vector **[I]**	Finnish (9)	12.60 EER, ∼69 CR
Attributes+i-vector **[IV]**	Finnish (8)	9.21 EER, ∼72 CR
Attributes+i-vector **[IV]**	English (7)	11.09 EER, ∼70 CR

attribute system outperforms the spectral i-vector system on 7 out of 8 accents with a statistical significance level of 5%.

[V]: The proposed OOS selection method outperforms the classical one-class SVM by a 16% relative reduction in EER. Furthermore, integrating the proposed method into the open-set LID task yields a 15% relative reduction in the detection cost in comparison to treating all of the development data as additional data in order to train the OOS model.

Comparison with other studies

Table 7.2 summarizes several previous studies on foreign accent recognition. As previously mentioned, there are several different forms of accents. However, this thesis focuses on *foreign* accent, namely, the type of language variation which occurs when non-native speakers use the characteristics of their L1 in L2, and as such, the author has attempted to do a representative sampling of the available research literature in order to place the achieved results within the context of competing or state-of-the-art approaches. However, in Table 7.2, due to the differences in datasets and evaluation metrics, the achieved foreign accent recognition results are not directly comparable with the literature. Thus, the evaluation numbers should be cautiously compared.

Comparing classification rate (CR) and EER numbers in **[IV]** and **[I]** with [149], one may note that they are reasonably simi-

lar. Similarly, CR numbers in [**IV**] and [**I**] are comparable with the CR reported in [150]. In a number of recent studies [113, 151], authors have integrated phonetic vowel information into conventional GMM-UBM framework in order to discriminate between 7 foreign accented English speeches. While, the authors have adopted a phone recognizer in order to extract vowels from speech, their findings, in demonstrating the usefulness of phonetic knowledge in order to characterize foreign accents in an acoustic-based framework, are in line with the findings of this thesis.

8 Conclusion

This thesis focused on the problem of automatic foreign accent recognition, that is, the task of identifying the mother tongue of a non-native speaker given a speech utterance spoken in his or her second language. Given that attribute features were used earlier in speech and language recognition tasks, this dissertation has significantly contributed to the adaptation of such methodology to the task of automatic foreign accent recognition. Using attribute features, we proposed a unified acoustic and phonotactically-inspired approach which benefits from both acoustic and phonotactic knowledge in order to discriminate between foreign spoken accents. We also proposed an i-vector representation framework to model the attribute feature streams. The key findings achieved in this work can be summarized as follows:

1. Incorporating phonotactically-inspired knowledge to the state-of-the-art spectral feature extraction methods, in the form of attribute features, significantly improves foreign accent recognition accuracy.

2. Appending temporal context in the forms of delta and double delta features to the attribute feature streams, substantially improves foreign accent recognition accuracy. Further, by applying a PCA over the temporal window of several frames, we were able to incorporate longer context size into the attribute feature streams, resulting in the best recognition results.

3. Training the system's hyper-parameters, such as universal background model, total variability matrix and dimensionality reduction projection matrices, from non-matched datasets yields poor recognition accuracy in comparison to training them from application-specific dataset with matched language.

4. In regards to an analysis of the factors affecting foreign accent recognition results, older speaker groups show lower detection error rates than younger ones, suggesting that mother tongue traits might be more preserved in older speakers when speaking L2 than younger speakers. Further, detecting foreign accents from those who have higher language proficiency in L2 are found to be more difficult than those with lower L2 language proficiency grades. Where (or how) an L2 is spoken, i.e. in university, at work, at home or as a hobby, does not have a considerable effect on the detection error rates.

5. Analyzing the effects of training set size and test utterance length on the overall foreign accent recognition performance suggests that the proposed attribute-based foreign accent recognition system outperforms the spectral system in all of the tested cases.

6. Selecting the most representative OOS data from a large set of unlabeled data to model OOS classes results in higher identification accuracy compared with pooling out of the entire unlabeled data to train an additional OOS class in an open-set LID task. Integrating our proposed OOS selection method into an open-set LID task relatively decreases the identification cost.

It should also be noted that the foreign accent recognition results reported in this dissertation were achieved on the two selected L2 languages, *Finnish* and *English*. The system developed in this thesis can be used for foreign accent recognition in other languages although similar results may not be achieved. One of the limitations of the proposed attribute-based foreign accent recognition system is that the bank of speech attribute detectors ignore the correlation between the attribute classes which might be useful for foreign accent discrimination.

Regarding future work, there is still room for improvement in the attribute based detectors. In particular, alternative acoustic features would be desirable to investigate for the training of the neural

network attribute detectors. Furthermore, with a renewed interest in DNN and their superior performance in language and speaker recognition tasks, it might be beneficial to adopt similar approaches to our foreign accent recognition tasks, specifically, in order to improve attribute detector accuracy [152]. Further, the performance of the proposed foreign accent recognition approach is investigated for extremely short utterances (i.e. less than 3 seconds). Finally, to reduce time complexity of the proposed OOS data selection method in large datasets, a random sampling of the data is a potential solution which can be investigated.

References

[1] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics* (Springer, 2007).

[2] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic Speech recognition and Speech Variability: A Review," *Speech Communication* **49,** 763–786 (2007).

[3] T. Kinnunen and H. Li, "An Overview of Text-Independent Speaker Recognition: From Features to Supervectors," *Speech Communication* **52,** 12–40 (2010).

[4] H. Li, B. Ma, and K. A. Lee, "Spoken Language Recognition: From Fundamentals to Practice," *Proceedings of the IEEE* **101,** 1136–1159 (2013).

[5] P. H. Ptacek and E. K. Sander, "Age Recognition from Voice," *Journal of Speech, Language, and Hearing Research* **9,** 273–277 (1966).

[6] T. Bocklet, A. Maier, J. G. Bauer, F. Burkhardt, and E. Nöth, "Age and Gender Recognition for Telephone Applications Based on GMM Supervectors and Support Vector Machines," in *Proc. of ICASSP* (2008), pp. 1605–1608.

[7] S. G. Koolagudi and K. S. Rao, "Emotion Recognition from Speech: A Review," *International Journal of Speech Technology* **15,** 99–117 (2012).

[8] V. Fromkin, R. Rodman, and N. Hyams, *An Introduction to Language* (Cengage Learning, 2013).

[9] Lewis, M. Paul (editor), *Ethnologue: Languages of the World*, 16 ed. (SIL International, 2009).

[10] M. Ruhlen, *On the Origin of Languages: Studies in Linguistic Taxonomy* (Stanford University Press, 1994).

[11] J. Siegel, *Second Dialect Acquisition* (Cambridge University Press, 2010).

[12] N. Kartushina and U. H. Frauenfelder, "On the Role of L1 Speech Production in L2 Perception: Evidence from Spanish Learners of French," in *Proc. of INTERSPEECH* (2013), pp. 2118–2122.

[13] H. S. Magen, "The Perception of Foreign-Accented Speech," *Journal of Phonetics* **26,** 381–400 (1998).

[14] P. Hrubeš, "Some Problems with the Pronunciation of English Typical of Native Speakers of German (A Tentative Case Study)," (B.Sc. Thesis, Masaryk University, 2008).

[15] H. Franco, L. Neumeyer, V. Digalakis, and O. Ronen, "Combination of Machine Scores for Automatic Grading of Pronunciation quality," *Speech Communication* **30,** 121–130 (2000).

[16] M. Eskenazi, "Detection of Foreign Speakers' Pronunciation Errors for Second Language Training - Preliminary Results," in *Proc. of ICSLP* (1996), pp. 1465–1468.

[17] D. Birch and J. McPhail, "The Impact of Accented Speech in International Television Advertisements," *Global Business Languages* **2,** 91–105 (2010).

[18] *Border Security: Fraud Risks Complicate State's Ability to Manage Diversity Visa Program* (United States Government Accountability Office, 2007).

[19] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition* (Prentice Hall, 1993).

[20] K. Bartkova and D. Jouvet, "Automatic Detection of Foreign Accent for Automatic Speech Recognition," in *Proc. of ICPHS* (2007), pp. 2185–2188.

[21] A. D. Lawson, D. M. Harris, and J. J. Grieco, "Effect of Foreign Accent on Speech Recognition in the NATO N-4 Corpus," in *Proc. of EUROSPEECH* (2003), pp. 1505–1508.

[22] Z. Wang, T. Schultz, and A. Waibel, "Comparison of Acoustic Model Adaptation Techniques on Non-native Speech," in *Proc. of ICASSP* (2003), pp. 540–543.

[23] C. Huang, T. Chen, and E. Chang, "Accent Issues in Large Vocabulary Continuous Speech Recognition," *International Journal of Speech Technology* **7,** 141–153 (2004).

[24] C. Huang, T. Chen, S. Li, E. Chang, and J. Zhou, "Analysis of Speaker Variability," in *Proc. of EUROSPEECH* (2001), pp. 1377–1380.

[25] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent Detection and Speech Recognition for Shanghai-Accented Mandarin," in *Proc. of INTERSPEECH* (2005), pp. 217–220.

[26] M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition," *Foundations and Trends in Signal Processing* **1,** 195–304 (2008).

[27] J.-L. Gauvain and C.-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio Processing* **2,** 291–298 (1994).

[28] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-Accent Deep Neural Network Acoustic Model with Accent-Specific Top Layer Using the KLD-Regularized Model Adaptation," in *Proc. of INTERSPEECH* (2014), pp. 2977–2981.

[29] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks," in *Proc. of ICLR* (2013).

[30] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing* **28,** 357–366 (1980).

[31] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data," *IEEE Intelligent Systems* **24,** 8–12 (2009).

[32] J. H. L. Hansen and L. M. Arslan, "Foreign Accent Classification Using Source Generator Based Prosodic Features," in *Proc. of ICASSP* (1995), pp. 836–839.

[33] M.-J. Kolly, A. Leemann, and V. Dellwo, "Foreign Accent Recognition Based on Temporal Information Contained in Lowpass-filtered Speech," in *Proc. of INTERSPEECH* (2014), pp. 2175–2179.

[34] M.-J. Kolly and V. Dellwo, "Cues to Linguistic Origin: The Contribution of Speech Temporal Information to Foreign Accent Recognition," *Journal of Phonetics* **42,** 12–23 (2014).

[35] M. A. Zissman and K. M. Berkling, "Automatic Language Identification," *Speech Communication* **35,** 115–124 (2001).

[36] Y. Yan and E. Barnard, "An Approach to Automatic Language Identification Based on Language-Dependent Phone Recognition," in *Proc. of ICASSP* (1995), pp. 3511–3514.

[37] L. F. D'Haro, O. Glembek, O. Plchot, P. Matejka, M. Soufifar, R. Cordoba, and J. Černocký, "Phonotactic Language Recognition using i-Vectors and Phoneme Posteriogram Counts," in *Proc. of INTERSPEECH* (2012), pp. 42–45.

[38] M. A. Kohler and M. Kennedy, "Language Identification Using Shifted Delta Cepstra," in *Symposium on Circuits and Systems* (2002), pp. III–69–72.

[39] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language Recognition via i-Vectors and Dimen-

sionality Reduction," in *Proc. of INTERSPEECH* (2011), pp. 857–860.

[40] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Transactions on Speech and Audio Processing* **3,** 72–83 (1995).

[41] L. Rognoni and M. G. Busà, "Testing the Effects of Segmental and Suprasegmental Phonetic Cues in Foreign Accent Rating: An Experiment Using Prosody Transplantation," in *Proc. of New Sounds* (2014), pp. 547–560.

[42] K. Aoyama, J. E. Flege, S. G. Guion, R. Akahane-Yamada, and T. Yamada, "Perceived Phonetic Dissimilarity and L2 Speech Learning: the Case of Japanese /r/ and English /l/ and /r/," *Journal of Phonetics* **32,** 233–250 (2004).

[43] T. Riney and J. Anderson-Hsieh, "Japanese Pronunciation of English," *Japan Association for Language Teaching* **15,** 21–36 (1993).

[44] N. J. Lass, *Contemporary Issues in Experimental Phonetics* (Academic Press, 2012).

[45] F. Bian, "The Influence of Chinese Stress on English Pronunciation Teaching and Learning," *English Language Teaching* **6,** 199–211 (2013).

[46] M. Swan and B. Smith, *Learner English: A Teacher's Guide to Interference and Other Problems* (Cambridge University Press, 2001).

[47] T. Piske, I. R. A. MacKay, and J. E. Flege, "Factors Affecting Degree of Foreign Accent in an L2: A Review," *Journal of phonetics* **29,** 191–215 (2001).

[48] M. A. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Transation on Speech and Audio Processing* **4,** 31–44 (1996).

[49] A. Y. Ng and M. I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," in *Proc. of NIPS* (2001), pp. 841–848.

[50] C. Cortes and V. Vapnik, "Support-Vector Networks," *Maching Learning* **20,** 273–297 (1995).

[51] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing* **19,** 788–798 (2011).

[52] H. Sun, T. H. Nguyen, G. Wang, K. A. Lee, B. Ma, and H. Li, "I2R Submission to the 2015 NIST Language Recognition i-Vector Challenge," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2016), pp. 311–318.

[53] T. Stafylakis, P. Kenny, M. J. Alam, and M. Kockmann, "Speaker and Channel Factors in Text-Dependent Speaker Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24,** 65–78 (2016).

[54] M. D. Plumpe, T. F. Quatieri, and D. A. Reynolds, "Modeling of the Glottal Flow Derivative Waveform with Application to Speaker Identification," *IEEE Transactions on Speech and Audio Processing* **7,** 569–586 (1999).

[55] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **29,** 254–272 (1981).

[56] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing* (Springer, 2008).

[57] R. Vergin and D. O'Shaughnessy, "Pre-emphasis and Speech Recognition," in *Proc. of CCECE*, Vol. 2 (1995), pp. 1062–1065.

[58] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. of the IEEE* **66,** 51–83 (1978).

[59] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform* (Society for Industrial and Applied Mathematics, 1992).

[60] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, Version 3.4* (Cambridge University Engineering Department, 2006).

[61] W.-H. Chen, C. H. Smith, and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications* **25,** 1004–1009 (1977).

[62] S. Furui, "Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics," in *Proc. of ICASSP*, Vol. 11 (1986), pp. 1991–1994.

[63] H. Hermansky and N. Morgan, "RASTA Processing of Speech," *IEEE Transactions on Speech and Audio Processing* **2,** 578–589 (1994).

[64] O. Viikki, D. Bye, and K. Laurila, "A Recursive Feature Vector Normalization Approach for Robust Speech recognition in noise," in *Proc. of ICASSP* (1998), pp. 733–736.

[65] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing* **10,** 19–41 (2000).

[66] D. A. Reynolds, "Gaussian Mixture Models," *Encyclopedia of Biometrics, Second Edition* 827–832 (2015).

[67] 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 (Methodological)* **39,** 1–38 (1977).

[68] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

[69] A. Statnikov, C. F. Aliferis, D. P. Hardin, and I. Guyon, *A Gentle Introduction to Support Vector Machines in Biomedicine: Case Studies* (World Scientific Publishing Company, 2011).

[70] S. R. Gunn, "Support Vector Machines for Classification and Regression," *Technichal Report* (1998).

[71] K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, and B. Schölkopf, "An Introduction to Kernel-based Learning Algorithms," *IEEE Transactions on Neural Networks* **12,** 181–202 (2001).

[72] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, "Language Recognition with Support Vector Machines," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2004), pp. 285–288.

[73] J. A. Anderson and E. Rosenfeld, *Neurocomputing: Foundations of Research* (MIT Press, 1988).

[74] B. L. Kalman and S. C. Kwasny, "Why Tanh: Choosing a Sigmoidal Function," in *Proc. of IJCNN*, Vol. 4 (1992), pp. 578–581.

[75] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proc. of ICML* (2010), pp. 807–814.

[76] J. Han and C. Moraga, "The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning," in *Proc. of IWANN* (1995), pp. 195–201.

[77] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015).

[78] Q. V. Le, "A Tutorial on Deep Learning Part 1: Nonlinear Classifiers and The Backpropagation Algorithm," *Technichal Report* (2015).

[79] H. Suo, M. Li, P. Lu, and Y. Yan, "Using SVM as Back-end Classifier for Language Identification," *EURASIP Journal on Audio, Speech, and Music Processing* **2008,** 1–6 (2008).

[80] H. Suo, M. Li, T. Liu, P. Lu, and Y. Yan, "The Design of Back-end Classifiers in PPRLM System for Language Identification," in *Proc. of ICNC* (2007), pp. 678–682.

[81] M. A. Zissman and E. Singer, "Automatic Language Identification of Telephone Speech Messages Using Phoneme Recognition and N-gram Modeling," in *Proc. of ICASSP* (1994), pp. 305–308.

[82] Y. Deng, W. Zhang, Y. Qian, and J. Liu, "Integration of Complementary Phone Recognizers for Phonotactic Language Recognition," in *Proc. of ICICA* (2010), pp. 237–244.

[83] M. K. Ravishankar, "Efficient Algorithms for Speech Recognition," (PhD Thesis, Carnegie Mellon University, 1996).

[84] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language Identification: A Tutorial," *IEEE Circuits and Systems Magazine* **11,** 82–108 (2011).

[85] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Prentice Hall Series in Artificial Intelligence, 2000).

[86] M. A. Zissman, "Predicting, Diagnosing and Improving Automatic Language Identification Performance," in *Proc. of EUROSPEECH* (1997), pp. 51–54.

[87] Q. Zhang and J. H. L. Hansen, "Training Candidate Selection for Effective Rejection in Open-set Language Identification," in *Proc. of SLT* (2014), pp. 384–389.

[88] NIST, "The 2015 Language Recognition i-Vector Machine Learning Challenge," *Technichal Report* (2015).

[89] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods* (Cambridge University Press, 2000).

[90] J. Zhang, "Detecting Outlying Subspaces for High-dimensional Data: A Heuristic Search Approach," in *Proc. of SIAM* (2005), pp. 80–86.

[91] N. V. Smirnov, "Estimate of Deviation Between Empirical Distribution Functions in Two Independent Samples," *Bulletin Moscow University* **2,** 3–16 (1939).

[92] M. S. Kim, "Robust, Scalable Anomaly Detection for Large Collections of Images," in *Proc. of SocialCom* (2013), pp. 1054–1058.

[93] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference* (CRC Press, 2011).

[94] P. Kenny and P. Dumouchel, "Experiments in Speaker Verification Using Factor Analysis Likelihood Ratios," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2004), pp. 219–226.

[95] S. E. Shepstone, K. A. Lee, H. Li, Z.-H. Tan, and S. H. Jensen, "Total Variability Modeling Using Source-Specific Priors," *IEEE/ACM Transactions on Audio, Speech and Language Processing* **24,** 504–517 (2016).

[96] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice Modeling with Sparse Training Data," *IEEE Transactions on Speech and Audio Processing* **13,** 345–354 (2005).

[97] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Inter-Speaker Variability in Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing* **16,** 980–988 (2008).

[98] B. Desplanques, K. Demuynck, and J.-P. Martens, "Combining Joint Factor Analysis and i-Vectors for Robust Language Recognition," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2014), pp. 73–80.

[99] F. Verdet, D. Matrouf, J.-F. Bonastre, and J. Hennebert, "Factor Analysis and SVM for Language Recognition," in *Proc. of INTERSPEECH* (2009), pp. 164–167.

[100] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-Class Covariance Normalization for SVM-based Speaker Recognition," in *Proc. of INTERSPEECH* (2006), pp. 1471–1474.

[101] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition* (John Wiley & Sons, 2004).

[102] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proc. of ICCV* (2007), pp. 1–8.

[103] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation," in *Proc. of ICASSP* (2006), pp. 97–100.

[104] Y. Xu, I. V. McLoughlin, Y. Song, and K. Wu, "Improved i-Vector Representation for Speaker Diarization," *Circuits, Systems, and Signal Processing* **35,** 3393–3404 (2016).

[105] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "i-Vector Approach to Phonotactic Language Recognition," in *Proc. of INTERSPEECH* (2011), pp. 2913–2916.

[106] H. Hotelling, "Analysis of a Complex of Statistical Variables with Principal Components," *Journal of Educational Psychology* **24,** 417–441 (1933).

[107] I. T. Jolliffe, *Principal Component Analysis* (Springer, 1986).

[108] R. P. W. Duin and M. Loog, "Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26,** 732–739 (2004).

[109] N. Kumar, "Investigation of Silicon Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition," (PhD Thesis, Johns Hopkins University, 1997).

[110] L. Burget, "Combination of Speech Features Using Smoothed Heteroscedastic Linear Discriminant Analysis," in *Proc. of INTERSPEECH* (2004), pp. 2549–2552.

[111] L. Burget, P. Matějka, P. Schwarz, O. Glembek, and J. Černocký, "Analysis of Feature Extraction and Channel Compensation in GMM Speaker Recognition System," *IEEE Transactions on Audio, Speech, and Language Processing* **15,** 1979–1986 (2007).

[112] M. Rouvier, R. Dufour, G. Linarès, and Y. Estève, "A Language-Identification Inspired Method for Spontaneous Speech Detection," in *Proc. of INTERSPEECH* (2010), pp. 1149–1152.

[113] Z. Ge, "Improved Accent Classification Combining Phonetic Vowels with Acoustic Features," in *Proc. of CISP* (2015), pp. 1204–1209.

[114] M. J. F. Gales, "Semi-Tied Covariance Matrices For Hidden Markov Models," *IEEE Transactions on Speech and Audio Processing* **7,** 272–281 (1999).

[115] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language Recognition in i-Vectors Space," in *Proc. of INTERSPEECH* (2011), pp. 861–864.

[116] A. H. Poorjam, R. Saeidi, T. Kinnunen, and V. Hautamäki, "Incorporating Uncertainty as a Quality Measure in i-Vector

Based Language Recognition," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2016), pp. 74–80.

[117] S. Schmid, "The Pronunciation of Voiced Obstruents in L2 French: A Preliminary Study of Swiss German Learners," *Poznan Studies in Contemporary Linguistics* **48,** 627–659 (2012).

[118] C.-A. Forel and G. Puskás, *Phonetics and Phonology: Reader for First Year English Linguistics* (University of Geneva, 2005).

[119] V. H. Do, X. Xiao, E. S. Chng, and H. Li, "Context-Dependent Phone Mapping for Acoustic Modeling of Under-resourced Languages," *International Journal of Asian Language Processing* **23,** 21–33 (2015).

[120] S. Siniscalchi and C.-H. Lee, "An Attribute Detection Based Approach to Automatic Speech Processing," *Loquens* **1** (2014).

[121] H. Alroqi, *LANE 321 Introduction to linguistics lecture notes* (King Abdulaziz University, 2015).

[122] C.-H. Lee, M. A. Clements, S. Dusan, E. Fosler-Lussier, K. Johnson, B.-H. Juang, and L. R. Rabiner, "An Overview on Automatic Speech Attribute Transcription (ASAT)," in *Proc. of INTERSPEECH* (2007), pp. 1825–1828.

[123] C.-H. Lee and S. M. Siniscalchi, "An Information-Extraction Approach to Speech Processing: Analysis, Detection, Verification, and Recognition," *Proceedings of the IEEE* **101,** 1089–1115 (2013).

[124] S. M. Siniscalchi, T. Svendsen, and C.-H. Lee, "Towards Bottom-up Continuous Phone Recognition," in *IEEE Workshop on Automatic Speech Recognition Understanding* (2007), pp. 566–569.

[125] S. M. Siniscalchi, T. Svendsen, and C.-H. Lee, "Toward a Detector-based Universal Phone Recognizer," in *Proc. of ICASSP* (2008), pp. 4261–4264.

[126] H. Li, B. Ma, and C.-H. Lee, "A Vector Space Modeling Approach to Spoken Language Identification," *IEEE Transactions on Audio, Speech and Language Processing* **15,** 271–284 (2007).

[127] P. Avery and S. Ehrlich, *Teaching American English Pronunciation* (Oxford University Press, 2013).

[128] L. T. Tuan, "Vietnamese EFL Learners' Difficulties with English Consonants," *Studies in Literature and Language* **3,** 56–67 (2011).

[129] N. T. Chi, "Techniques to Improve English Pronunciation for Second-Major Students at Hai Phong Private University," (PhD Thesis, Hai Phong Private University, 2009).

[130] I. Thompson, "Foreign Accents Revisited: The English Pronunciation of Russian Immigrants," *Language Learning* **41,** 177–204 (1991).

[131] A. J. Sewell, "Phonological Features of Hong Kong English: Patterns of Variation and Effects on Local Acceptability," (PhD Thesis, Lingnan University, 2010).

[132] C.-H. Lee, "From Knowledge-Ignorant to Knowledge-Rich Modeling: A New Speech Research Paradigm for Next Generation Automatic Speech Recognition," in *Proc. of INTERSPEECH* (2004), pp. 109–112.

[133] S. M. Siniscalchi and C.-H. Lee, "A Study on Integrating Acoustic-Phonetic Information into Lattice Rescoring for Automatic Speech Recognition," *Speech Communication* **51,** 1139–1153 (2009).

[134] K. Kumpf and R. W. King, "Automatic Accent Classification of Foreign Accented Australian English Speech," in *Proc. of ICSLP* (1996), pp. 1740–1743.

[135] E. Ahn, "A Computational Approach to Foreign Accent Classification," (2016), Honors Thesis Collection. Paper 323.

[136] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-dimensional Total Variability Space for Speaker Verication," in *Proc. of INTERSPEECH* (2009), pp. 1559–1562.

[137] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on Cross-Language Attribute Detection and Phone Recognition With Minimal Target-Specific Training Data," *IEEE Transations of Audio, Speech and Language Processing* **20,** 875–887 (2012).

[138] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika, "The OGI Multi-Language Telephone Speech Corpus," in *Proc. of ICSLP* (1992), pp. 895–898.

[139] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *Linguistic Data Consortium, Philadelphia* **33** (1993).

[140] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Phonotactic Language Recognition Using MLP Features," in *Proc. of INTERSPEECH* (2012), pp. 2041–2044.

[141] P. A. Torres-Carrasquillo, T. P. Gleason, and D. A. Reynolds, "Dialect Identification Using Gaussian Mixture Models," in *Proc. of The Speaker and Language Recognition Workshop (Odyssey)* (2004), pp. 757–760.

[142] NIST, "The 2015 NIST Language Recognition Evaluation Plan (LRE15)," *Technichal Report* (2015).

[143] N. Brümmer, "Focal Multi-class: Toolkit for Evaluation, Fusion and Calibration of Multi-class Recognition Scores," (online), available: `https://sites.google.com/site/nikobrummer/focalmulticlass`.

[144] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detec-

tion Task Performance," in *Proc. of EUROSPEECH* (1997), pp. 1895–1898.

[145] "Finnish National Foreign Language Certificate Corpus," (online), available: `http://yki-korpus.jyu.fi`.

[146] NIST, "The NIST Year 2008 Speaker Recognition Evaluation Plan," *Technichal Report* (2008).

[147] "Satakunta in Speech - the Current Finnish Dialects in the Area of Satakunta," (online), available: `www.utu.fi/en/units/hum/units/finnishandfinnougric/research/projects/Pages/Satankuntainthespeech.aspx`.

[148] P. Angkititraku and J. H. L. Hansen, "Advances in Phone-based Modeling for Automatic Accent Classification," in *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14 (2006), pp. 634–646.

[149] M. K. Omar and J. Pelecanos, "A Novel Approach to Detecting Non-native Speakers and Their Native Language," in *Proc. of ICASSP* (2010), pp. 4398–4401.

[150] M. H. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent Recognition Using i-Vector, Gaussian Mean Supervector, Guassian Posterior Probability for Spontaneous Telephone Speech," in *Proc. of ICASSP* (2013), pp. 7344–7348.

[151] Z. Ge, Y. Tan, and A. Ganapathiraju, "Accent Classification with Phonetic Vowel Representation," in *Proc. of ACPR* (2015), pp. 529–533.

[152] V. Hautamäki, S. M. Siniscalchi, H. Behravan, V. M. Salerno, and I. Kukanov, "Boosting Universal Speech Attributes Classification with Deep Neural Network for Foreign Accent Characterization," in *Proc. of INTERSPEECH* (2015), pp. 408–412.

Paper I

H. Behravan, V. Hautamäki, and T. Kinnunen
"Factors Affecting i-Vector Based Foreign Accent Recognition: A Case Study in Spoken Finnish"
Speech Communication,
66, 118–129 (2015).

Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 66 (2015) 118–129

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Factors affecting i-vector based foreign accent recognition: A case study in spoken Finnish

Hamid Behravan [a,b,*], Ville Hautamäki [a], Tomi Kinnunen [a]

[a] School of Computing, University of Eastern Finland, Box 111, FIN-80101 Joensuu, Finland
[b] School of Languages and Translation Studies, University of Turku, Turku, Finland

Received 22 December 2013; received in revised form 19 September 2014; accepted 15 October 2014
Available online 23 October 2014

Abstract

i-Vector based recognition is a well-established technique in state-of-the-art speaker and language recognition but its use in dialect and accent classification has received less attention. In this work, we extensively experiment with the spectral feature based i-vector system on Finnish foreign accent recognition task. Parameters of the system are initially tuned with the CallFriend corpus. Then the optimized system is applied to the *Finnish national foreign language certificate* (FSD) corpus. The availability of suitable Finnish language corpora to estimate the hyper-parameters is necessarily limited in comparison to major languages such as English. In addition, it is not immediately clear which factors affect the foreign accent detection performance most. To this end, we assess the effect of three different components of the foreign accent recognition: (1) recognition system parameters, (2) data used for estimating hyper-parameters and (3) language aspects. We find out that training the hyper-parameters from non-matched dataset yields poor detection error rates in comparison to training from application-specific dataset. We also observed that, the mother tongue of speakers with higher proficiency in Finnish are more difficult to detect than of those speakers with lower proficiency. Analysis on age factor suggests that mother tongue detection in older speaker groups is easier than in younger speaker groups. This suggests that mother tongue traits might be more preserved in older speakers when speaking the second language in comparison to younger speakers.

Keywords: Foreign accent recognition; i-Vector; Language proficiency; Age of entry; Level of education; Where second language is spoken

1. Introduction

Foreign spoken accents are caused by the influence of one's first language on the second language (Flege et al., 2003). For example, an English–Finnish bilingual speaker may have an English accent in his/her spoken Finnish because of learning Finnish later in life. Non-native speakers induce variations in different word pronunciation and grammatical structures into the second language (Grosjean, 2010). Interestingly, these variations are not random across speakers of a given language, because the original mother tongue is the source of these variations (Witteman, 2013). Nevertheless, between-speaker differences, gender, age and anatomical differences in vocal tract generate within-language variation (Witteman, 2013). These variations are nuisance factors that adversely affect detection of the mother tongue.

Foreign accent recognition is a topic of great interest in the areas of intelligence and security including immigration and border control sites. It may help officials to detect travelers with a fake passport by recognizing the immigrant's actual country and region of spoken foreign accent (GAO, 2007). It has also a wide range of commercial

* Corresponding author at: School of Computing, University of Eastern Finland, Box 111, FIN-80101 Joensuu, Finland.
E-mail addresses: behravan@cs.uef.fi (H. Behravan), villeh@cs.uef.fi (V. Hautamäki), tkinnu@cs.uef.fi (T. Kinnunen).

http://dx.doi.org/10.1016/j.specom.2014.10.004
0167-6393/

applications including services based on user-agent voice commands and targeted advertisement.

Similar to spoken language recognition (Li et al., 2013), various techniques including *phonotactic* (Kumpf and King, 1997; Wu et al., 2010) and *acoustic* approaches (Bahari et al., 2013; Scharenborg et al., 2012; Behravan et al., 2013) have been proposed to solve the foreign accent detection task. The former uses phonemes and phone distributions to discriminate different accents; in practice, it uses multiple phone recognizer outputs followed by language modeling (Zissman, 1996). The acoustic approach in turn uses information taken directly from the spectral characteristics of the audio signals in the form of *mel-frequency cepstral coefficient* (MFCC) or *shifted delta cepstra* (SDC) features derived from MFCCs (Kohler and Kennedy, 2002). The spectral features are then modeled by a "bag-of-frames" approach such as *universal background model* (UBM) with adaptation (Torres-Carrasquillo et al., 2004) and *joint factor analysis* (JFA) (Kenny, 2005). For an excellent recent review of the current trends and computational aspects involved in general language recognition tasks including foreign accent recognition, we point the interested reader to (Li et al., 2013).

Among the acoustic systems, total variability model or *i-vector* approach originally used for speaker recognition (Dehak et al., 2011a), has been successfully applied to language recognition tasks (González et al., 2011; Dehak et al., 2011b). It consists of mapping speaker and channel variabilities to a low-dimensional space called *total variability space*. To compensate intersession effects, this technique is usually combined with *linear discriminant analysis* (LDA) (Fukunaga, 1990) and *within-class covariance normalization* (WCCN) (Kanagasundaram et al., 2011).

The i-vector approach has received less attention in dialect and accent recognition systems. Caused by more subtle linguistic variations, dialect and accent recognition are generally more difficult than language recognition (Chen et al., 2010). Thus, it is not obvious how well i-vectors will perform on these tasks. However, more fundamentally, the i-vector system has many data-driven components for which training data needs to be selected. It would be tempting to train some of the hyper-parameters on a completely different out-of-set-data (even different language), and leave only the final parts – training and testing a certain dialect or accent – to the trainable parts. This is also motivated by the fact that there is a lack of linguistic resources available for languages like Finnish, comparing to English for which corpora from NIST[1] and LDC[2] exist.

The i-vector based dialect and accent recognition has previously been addressed in (DeMarco and Cox, 2012; Bahari et al., 2013). DeMarco and Cox (2012) addressed a British dialect classification task with fourteen dialects, resulting in 68% overall classification rate while (Bahari et al., 2013) compared three accent modeling approaches in classifying English utterances produced by speakers of seven different native languages. The accuracy of the i-vector system was found comparable as compared to the other two existing methods. These studies indicate that the i-vector approach is promising for dialect and foreign accent recognition tasks. However, it can be partly attributed to availability of massive development corpora including thousands of hours of spoken English utterances to train all the system hyper-parameters. The present study presents a case when such resources are not available.

Comparing with the prior studies including our own preliminary analysis (Behravan et al., 2013), the new contribution of this study is a detailed account into factors affecting the i-vector based foreign accent detection. We study this from three different perspectives: parameters, development data, and language aspects. Firstly, we study how the various i-vector extractor **parameters**, such as the UBM size and i-vector dimensionality, affect accent detection accuracy. This classifier optimization step is carried out using the speech data from the CallFriend corpus (Canavan and Zipperle, 1996). As a minor methodological novelty, we study applicability of *heteroscedastic linear discriminant analysis* (HLDA) for supervised dimensionality reduction of i-vectors. Secondly, we study **data**-related questions on our accented Finnish language corpus. We explore how the choices of the development data for UBM, i-vector extractor and HLDA matrices affect accuracy; we study whether these could be trained using a different language (English). if the answer turn out positive, the i-vector approach would be easy to adopt to other languages without recourse to the computationally demanding steps of UBM and i-vector extractor training. Finally, we study **language aspects**. This includes three analyses: ranking of the original accents in terms of their detection difficulty, study of confusion patterns across different accents and finally, relating recognition accuracy with four affecting factors such as Finnish language proficiency, age of entry, level of education and where the second language is spoken.

Our hypothesis for the Finnish language proficiency is that recognition accuracy would be adversely affected by proficiency in Finnish. In other words, we expect higher accent detection errors for speakers who speak fluent Finnish. For the age of entry factor, we expect that the younger a speaker enters a foreign country, the higher the probability of fluency in the second language. Thus, we hypothesize that it is more difficult to detect the speaker's mother tongue in younger age groups than in older ones. This hypothesis is reasonable also because older people tend to keep their mother tongue traits more often than younger people (Munoz, 2010). Regarding the education factor, we hypothesize that mother tongue detection is more difficult in higher educated speakers than in lower educated ones. Finally, We also hypothesize that mother tongue detection is more difficult for the person who consistently use their second languages for social interaction

[1] http://www.itl.nist.gov/iad/mig/tests/spk/.
[2] http://www.ldc.upenn.edu/.

as compared to the speakers who do not use their second language in regular basis for social interaction.

2. System components

Fig. 1 shows the block diagram of the method used in this work. The i-vector system consists of two main part: front-end and back-end. The former consists of cepstral feature extraction and UBM training, whereas the latter includes sufficient statistics computation, training of the T-matrix, i-vector extraction, dimensionality reduction and scoring.

2.1. i-vector system

i-Vector modeling (Dehak et al., 2011a) is inspired by the success of *joint factor analysis* (JFA) (Kenny et al., 2008) in speaker verification. In JFA, speaker and channel effects are independently modeled using *eigenvoice* (speaker subspace) and *eigenchannel* (channel subspace) models:

$$\mathbf{M} = \mathbf{m} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x}, \tag{1}$$

where $\mathbf{M}$ is the speaker supervector, $\mathbf{m}$ is a speaker and channel independent supervector created by concatenating the centers of UBM and low-rank matrices $\mathbf{V}$ and $\mathbf{U}$ represent, respectively, linear subspaces for speaker and channel variability in the original mean supervector space. The latent variables $\mathbf{x}$ and $\mathbf{y}$ are assumed to be independent of each other and have a standard normal distributions, i.e. $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Dehak et al. (2011a) found that these subspaces are not completely independent, therefore a combined total variability modeling was introduced.

In the i-vector approach, the GMM supervector ($\mathbf{M}$) of each accent utterance is decomposed as (Dehak et al., 2011a),

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{2}$$

where $\mathbf{m}$ is again the UBM supervector, $\mathbf{T}$ is a low-rank rectangular matrix, representing between-utterance variability in the supervector space, and $\mathbf{w}$ is the i-vector, a standard normally distributed latent variable drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The $\mathbf{T}$ matrix is trained using a similar technique which is used train $\mathbf{V}$ in JFA, except that each training utterance of a speaker model is treated as belonging to different speakers. Therefore, in contrast to JFA, the $\mathbf{T}$ matrix training does not need speaker or dialect labels. To this end, i-vector approach is an unsupervised learning method. The i-vector $\mathbf{w}$ is estimated from its posterior distribution conditioned on the Baum–Welch statistics extracted from the utterance using the UBM (Dehak et al., 2011a).

The i-vector extraction can be seen as a mapping from a high-dimensional GMM supervector space to a low-dimensional i-vector that preserves most of the variability. In this work, we use 1000-dimensional that are further length normalized and whitened (Garcia-Romero and Espy-Wilson, 2011).

Cosine scoring is commonly used for measuring similarity of two i-vectors (Dehak et al., 2011a). The cosine score t of the test i-vector, $\mathbf{w}_{\text{test}}$, and the i-vectors of target accent a, $\mathbf{w}^a_{\text{target}}$, is defined as their inner product $\langle \mathbf{w}_{\text{test}}, \mathbf{w}^a_{\text{target}} \rangle$ and computed as follows:

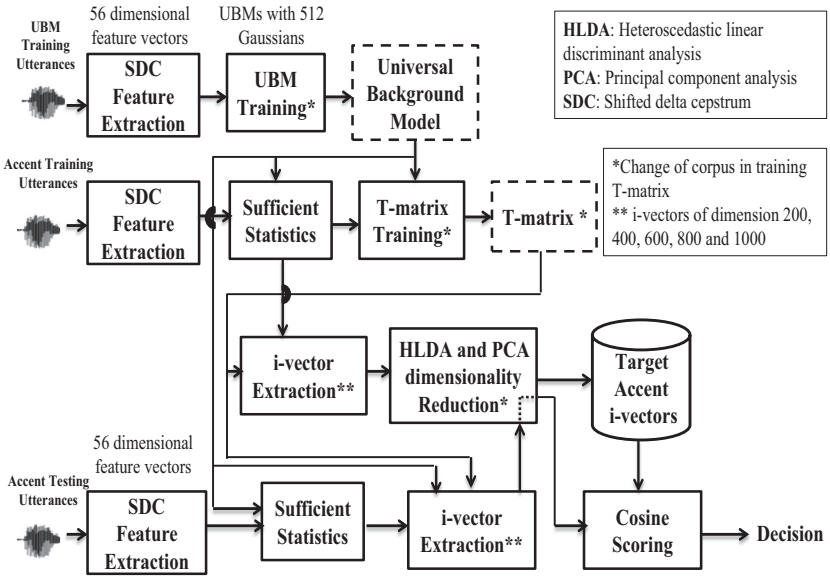


Fig. 1. The block diagram of the method used in this work.

$$t = \frac{\hat{\mathbf{w}}_{\text{test}}^{\text{T}}\, \hat{\mathbf{w}}_{\text{target}}^{a}}{\|\hat{\mathbf{w}}_{\text{test}}\|\ \|\hat{\mathbf{w}}_{\text{target}}^{a}\|}, \tag{3}$$

where $\hat{\mathbf{w}}_{\text{test}}$ is,

$$\hat{\mathbf{w}}_{\text{test}} = \mathbf{A}^{\text{T}}\mathbf{w}_{\text{test}}, \tag{4}$$

and $\mathbf{A}$ is the HLDA projection matrix (Loog and Duin, 2004) to be detailed below in Section 2.2. Further, $\hat{\mathbf{w}}_{\text{target}}^{a}$ is the average i-vector over all the training utterances in accent a, i.e.

$$\hat{\mathbf{w}}_{\text{target}}^{a} = \frac{1}{N_a}\sum_{i=1}^{N_a}\hat{\mathbf{w}}_i^a, \tag{5}$$

where N_a is the number of training utterances in accent a and $\hat{\mathbf{w}}_i^a$ is the projected i-vector of training utterance i from accent a, computed the same way as (4).

Obtaining the scores $\{t_a,\ a = 1,\ldots,L\}$ for a particular test utterance compared with all the L target accent models of accent a, those scores are further post-processed as (Brümmer and van Leeuwen, 2006):

$$t'(a) = \log\frac{\exp(t_a)}{\frac{1}{L-1}\sum_{k\neq a}\exp(t_k)}, \tag{6}$$

where $t'(a)$ is the detection log-likelihood ratio or final score used in the detection task.

2.2. Reducing the i-vector dimensionality

As the extracted i-vectors contain both intra- and between-accent variations, the aim of dimensionality reduction is to project the i-vectors onto a space where between-accent variability is maximized and intra-accent variability is minimized. Traditionally, LDA is used to perform dimensionality reduction where, for R-class classification problem, the maximum projected dimension is $R-1$.

As (Loog and Duin, 2004) argue, these $R-1$ dimensions do not necessarily contain all the discriminant information for the classification task. Moreover, LDA separates only the class means and it does not take into account the discriminatory information in the class covariances. In recent years, an extension of LDA, *heteroscedastic* linear discriminant analysis (HLDA), has gained popularity in speech research community. HLDA, unlike LDA, deals with discriminant information presented both in the means and covariance matrices of classes (Loog and Duin, 2004).

HLDA was originally introduced in (Kumar, 1997) for auditory feature extraction, and later applied to speaker (Burget et al., 2007) and language (Rouvier et al., 2010) recognition with the purpose of reducing dimensionality of GMM supervectors and acoustic features, respectively. In this work, we also use it to reduce the dimensionality of extracted i-vectors. For completeness, we briefly summarize the HLDA technique below.

In the HLDA technique, the i-vectors of dimension n are projected into first $p < n$ rows, $d_{j=1\ldots p}$, of $n \times n$ HLDA transformation matrix denoted by $\mathbf{A}$. The matrix $\mathbf{A}$ is estimated by an efficient row-by-row iteration method (Gales, 1999), whereby each row is iteratively estimated as,

$$\hat{\mathbf{d}}_k = \mathbf{c}_k\mathbf{G}^{k-1}\sqrt{\frac{N}{\mathbf{c}_k\mathbf{G}^{k-1}\mathbf{c}_k^{\text{T}}}}. \tag{7}$$

Here, $\mathbf{c}_k$ is the kth row vector of the co-factor matrix $\mathbf{C} = |\mathbf{A}|\,\mathbf{A}^{-1}$ for the current estimate of $\mathbf{A}$ and

$$\mathbf{G}^k = \begin{cases} \sum_{j=1}^{J}\frac{N_j}{\mathbf{d}_k\widehat{\boldsymbol{\Sigma}}^{(j)}\mathbf{d}_k^{\text{T}}}\widehat{\boldsymbol{\Sigma}}^{(j)} & k \leqslant p, \\ \frac{N}{\mathbf{d}_k\widehat{\boldsymbol{\Sigma}}\mathbf{d}_k^{\text{T}}}\widehat{\boldsymbol{\Sigma}} & k > p, \end{cases} \tag{8}$$

where $\widehat{\boldsymbol{\Sigma}}$ and $\widehat{\boldsymbol{\Sigma}}^{(j)}$ are estimates of the class-independent covariance matrix and the covariance matrix of the jth model, N_j is the number of training utterances of the jth model and N is the total number of training utterances. To avoid near-to-singular covariance matrices in HLDA training process, principal component analysis (PCA) is first applied (Loog and Duin, 2004; Rao and Mak, 2012) and the PCA-projected features are used as the inputs to HLDA. The dimension of PCA is selected in such a manner that most of the principal components are retained and within-models scatter matrix becomes non-singular (Loog and Duin, 2004).

2.3. Within-class covariance normalization

To compensate for unwanted intra-class variations in the total variability space, within-class covariance normalization (WCCN) (Hatch et al., 2006) is applied to the extracted i-vectors. To this end, a within-class covariance matrix, $\boldsymbol{\Lambda}$, is first computed using,

$$\boldsymbol{\Lambda} = \frac{1}{L}\sum_{a=1}^{L}\frac{1}{N_a}\sum_{i=1}^{N_a}(\mathbf{w}_i^a - \overline{\mathbf{w}}_a)(\mathbf{w}_i^a - \overline{\mathbf{w}}_a)^{\text{T}}, \tag{9}$$

where $\overline{\mathbf{w}}_a$ is the mean i-vector for each accent a, L is the number of target accents and N_a is the number of training utterances for the accent a. The inverse of $\boldsymbol{\Lambda}$ is then used to normalize the direction of the projected i-vectors in the cosine kernel. This is equivalent to projecting the i-vector subspace by the matrix $\mathbf{B}$ obtained by Cholesky decomposition of $\boldsymbol{\Lambda}^{-1} = \mathbf{B}\mathbf{B}^{\text{T}}$.

3. Experimental setup

3.1. Corpus

We use *Finnish national foreign language certificate* (FSD) corpus (University of Jyväskylä, 2000) to perform foreign accent classification task. The corpus consists of official language proficiency tests for foreigners interested in Finnish language proficiency certificate for the purpose of applying for a job or citizenship. All the data has been recorded by language experts. Generally, the test is intended for evaluating test-takers' proficiency in listening

Table 1
Grades within different levels in the FSD corpus.

Levels	Grades		
Basic	0	1	2
Intermediate	3	4	
Advanced	5	6	

comprehension, reading comprehension, speaking, and writing. This test can be taken at basic, intermediate and advanced levels. The test-takers choose the proficiency level at which they wish to participate. The difference between the levels is the extent and variety of expression required. At the basic level, it is important that test-takers convey their message in a basic form, while in the intermediate level, richer expression is required. More effective and natural expressions should be presented in the advanced level. However, communication purposes, i.e. functions and questions, are more or less the same at all levels. Table 1 shows the grading scale at each level of the tests in this corpus.[3]

For our purposes, we selected Finnish responses corresponding to 18 foreign accents. Unfortunately, as the number of utterances in some accents was not large enough, a limited number of eight accents – Russian, Albanian, Arabic, English, Estonian, Kurdish, Spanish, and Turkish – with enough data were chosen for the experiments. However, the unused accents were utilized in training the hyper-parameters of the i-vector system, the UBM and the T-matrix.

To perform the recognition task, each accent set is randomly partitioned into a training and a test subset. To avoid speaker and session bias, the same speaker was not placed into the test and train subsets. The test subset corresponds to (approximately) 40% of the utterances, while the training set corresponds to the remaining 60%. The original audio files, stored in MPEG-2 Audio Layer III (mp3) compressed format, were decompressed, resampled to 8 kHz and partitioned into 30-s chunks. Table 2 shows the distribution of train and test files in each target accent.

The NIST SRE 2004[4] corpus was chosen as the out-of-set-data for hyper-parameter training. For our purposes, 1000 gender-balanced utterances were randomly selected from this corpus to train the UBM and T-matrix. We note that this is an American English corpus of telephone-quality speech.

Unlike UBM and T-matrix, training the HLDA projection matrix requires labeled data. Since accent labels are not represented in the NIST corpus, we use the *CallFriend* corpus (Canavan and Zipperle, 1996) to train HLDA. This corpus is a collection of unscripted conversations of 12 languages recorded over telephone lines. It includes two dialects for each target language available. All utterances are organized into training, development and evaluation subsets. For our purposes, we selected all the training utterances from dialects of English, Mandarin and Spanish languages and partitioned them into 30-s chunks, resulting in approximately 4000 splits per each subset. All audio files have 8 kHz sampling rate.

Table 2
Train and test files distributions in each target accent in the FSD corpus.

Accent	No. of train files	No. of test files	No. of speakers
Spanish	47	25	15
Albanian	56	29	19
Kurdish	61	32	21
Turkish	66	34	22
English	70	36	23
Estonian	122	62	38
Arabic	128	66	42
Russian	556	211	235
Total	1149	495	415

3.2. Front-end configuration

The front-end consists of concatenation of MFCC and SDC coefficients (Kohler and Kennedy, 2002). To this end, speech signals framed with 20 ms Hamming window with 50% overlap are filtered by 27 mel-scale filters over 0–4000 Hz frequency range. RASTA filtering (Hermansky and Morgan, 1994) is applied to log-filterbank energies. Seven first cepstral coefficients (c0–c6) are computed using discrete cosine transform. The cepstral coefficients are further processed using utterance-level cepstral mean and variance normalization (CMVN) and vocal tract length normalization (VTLN) (Lee and Rose, 1996), and converted into 49-dimensional *shifted delta cepstra* (SDC) feature vectors with 7-1-3-7 configuration parameters (Kohler and Kennedy, 2002). These four parameters correspond to, respectively, the number of cepstral coefficients, time delay for delta computation, time shift between consecutive blocks, and number of blocks for delta coefficient concatenation. Removing non-speech frames, the 7 first MFCC coefficients (including c0) are further concatenated to SDCs to obtain 56-dimensional feature vectors.

In a preliminary experiment on our evaluation corpus FSD (Behravan, 2012), the combined feature set is shown to give a relative decrease in EER of more than 30% as compared to the only SDC feature based technique.

3.3. Objective evaluation metrics

System performance is reported in terms of both average *equal error rate* (EER_{avg}) and *average detection cost* (C_{avg}) (Li et al., 2013). EER indicates the operating point on detection error trade-off (DET) curve (Martin et al., 1997) at which false alarm and miss rates are equal. EER per target accent is computed in a manner that other accents serve as non-target trials. Average *equal error rate*

[3] The FSD corpus is available by request from http://yki-korpus.jyu.fi/. Filelists used in this study are available by request from the first author.
[4] http://catalog.ldc.upenn.edu/LDC2006S44.

(EER_{avg}) is computed by taking the average over all the L target accent EERs.

C_{avg}, in turn, is defined as follows (Li et al., 2013),

$$C_{avg} = \frac{1}{L}\sum_{a=1}^{L} C_{DET}(L_a), \tag{10}$$

where $C_{DET}(L_a)$ is the detection cost for subset of test segments trials for which the target accent is L_a:

$$C_{DET}(L_a) = C_{miss}P_{tar}P_{miss}(L_a) + C_{fa}(1-P_{tar}) \times \frac{1}{L-1}\sum_{m\neq a} P_{fa}(L_a, L_m). \tag{11}$$

P_{miss} denotes the miss probability (or false rejection rate), i.e. a test segment of accent L_a is rejected as not being in that accent. $P_{fa}(L_a, L_m)$ is the probability when a test segment of accent L_m is detected as accent L_a. It is computed for each target/non-target accent pairs. C_{miss} and C_{fa} are costs of making errors and are set to 1. P_{tar} is the prior probability of a target accent and is set to 0.5.

4. Results

We first optimize the i-vector parameters in the context of dialect and accent recognition tasks. For this purpose, we utilize the CallFriend corpus. The results are summarized in Table 3.

In Fig. 2, we show EER as a function of HLDA output dimension. We find that the optimal dimension of the HLDA projected i-vectors is 180 and too aggressive reduction in dimension decreases accuracy. We also find that accuracy improves with the increase of i-vector dimensionality as Table 4 shows. Furthermore, our results showed that the UBM with smaller size outperforms larger UBM as Table 5 shows. Based on these previous findings, UBM size, i-vector size and output dimensionality are set to 512, 1000 and 180, respectively.

4.1. Effect of development data on i-vector hyper-parameters estimation

Table 6 shows the results on the FSD corpus when the hyper-parameters are trained from different datasets. Here, WCCN and score normalization are not applied. By considering the first row with matched language as a baseline (13.37% EER_{avg}), we observe the impact of each of the hyper-parameter training configurations as follows:

Table 3
The i-vector system's optimum parameters as reported in (Behravan et al., 2013).

i-vector parameters	Search range and **optima**
UBM size	**256**, **512**, 1024, 2048, 4096
i-vector dimensionality	200, 400, 600, 800, **1000**
HLDA dimensionality	50, 100, 150, **180**, 220, 300, 350, 400

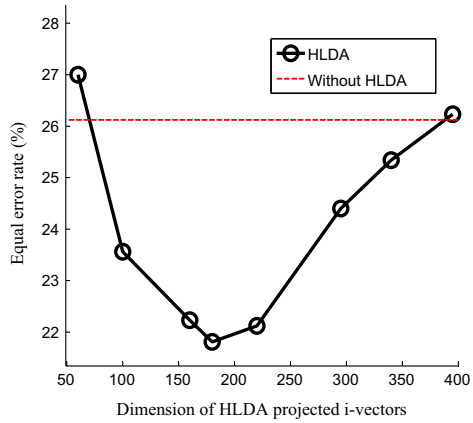


Fig. 2. Equal error rates at different dimensions of the HLDA projected i-vectors in the CallFriend corpus as reported in (Behravan et al., 2013).

- Effect of HLDA (row 1 vs row 2): EER_{avg} increases to 18.28% (relative increase of 37%).
- Effect of T-matrix (row 1 vs 3): EER_{avg} increases to 20.98% (relative increase of 57%).
- Effect of UBM (row 1 vs 4): EER_{avg} increases to 23.85% (relative increase of 78%).
- Effect of UBM and T-matrix (row 1 vs 5): EER_{avg} increases to 26.76% (relative increase of 101%).

In the light of these findings, it seems clear that the 'early' system hyper-parameters (UBM and T-matrix) have a much larger role and they should be trained from as closely matched data as possible; we see that when all the hyper-parameters are trained from the FSD corpus, the highest accuracy is achieved. The most severe degradation (101%) is attributed to the joint effect of UBM and T-matrix and the least severe (37%) to HLDA, T-matrix (57%) and UBM (78%) falling in between. It is instructive to recall the order of computations: sufficient statistics from UBM → i-vector extractor training → HLDA training. Since all the remaining steps depend on the "bottleneck" components, i.e. UBM and T-matrix, it is not surprising that they have the largest relative effect.

The generally large degradation relative to the baseline set-up with matched data is reasonably explained by the

Table 4
Performance of the i-vector system in the CallFriend corpus for selected i-vector dimensions (EER in %, form). UBM has 1024 Gaussians as reported in (Behravan et al., 2013).

i-vector dim.	English	Mandarin	Spanish
200	23.20	20.49	20.87
400	22.60	19.11	20.21
600	21.30	18.45	19.63
800	19.83	16.31	18.63
1000	**18.01**	**14.91**	**16.01**

Table 5
Performance of the i-vector system in the CallFriend corpus for five selected UBM sizes (EER in %, form). i-vectors are of dimension 600 as reported in (Behravan et al., 2013).

UBM size	English	Mandarin	Spanish
256	**21.12**	17.93	**19.00**
512	21.61	**17.91**	19.15
1024	21.30	18.45	19.63
2048	23.81	21.15	22.01
4096	23.89	21.57	22.66

Table 6
EER_{avg} and $C_{avg} \times 100$ performance for effect of changing datasets in training the i-vector hyper-parameters. (WCCN and score normalization turned off.)

UBM	T matrix	HLDA	EER_{avg}%	$C_{avg} \times 100$	Id_{error}%
Database used for training					
FSD	FSD	FSD	**13.37**	**7.04**	**33.65**
FSD	FSD	CallFriend	18.28	7.49	38.29
FSD	NIST	FSD	20.98	7.83	40.30
NIST	FSD	FSD	23.85	8.15	42.91
NIST	NIST	FSD	26.76	8.41	44.67

Table 7
Effect of score normalization on the recognition performance. (HLDA and WCCN turned on and off, respectively.)

Score normalization	EER_{avg}%	$C_{avg} \times 100$	Id_{error}%
No	13.37	7.04	33.65
Yes	**13.01**	**6.94**	**32.85**

large differences between type of data of evaluation corpus (FSD) and hyper-parameter estimation corpora (NIST SRE and CallFriend). FSD consists of Finnish language data recorded with close-talking microphones in a classroom environment. Even though speech is very clear, background babble noise from the other students is evident in all the recordings. This is contrast to the NIST SRE and CallFriend corpora where most of the speech files are recorded over telephone line and babble noise is less common.

The results of Table 6 were computed with WCCN and score normalization turned off. Let us now turn our attention to these additional system components. Firstly, Table 7 shows the effect of score normalization when all the hyper-parameters are trained from the FSD corpus (i.e., row 1 of Table 6). EER_{avg} decreases from 13.37% to 13.01%, which indicates a slightly increased recognition accuracy when the scores are normalized in the backend.

Secondly, Table 8 shows the joint effect of WCCN and HLDA on the recognition performance when all the hyper-parameters are trained from the FSD corpus (i.e., row 1 of Table 6). In addition to that, score normalization is also applied. EER_{avg} decreases from 17.10% to 12.60% when both HLDA and WCCN are applied. The worst case is when HLDA is turned off and WCCN is turned on. This is because turning off HLDA leads to inaccurate estimation of covariance matrix in higher dimensional i-vector space.

4.2. Comparing i-vector and GMM-UBM systems

In order to have a baseline comparison between the i-vector approach and the classical accent recognition systems, we used conventional GMM-UBM system with MAP adaptation similar to the work presented in (Torres-Carrasquillo et al., 2004). GMM-UBM system is simpler and computationally more efficient in comparison to the i-vector systems. Map adaptation consists of single iteration for adapting the UBM to each dialect model using SDC + MFCC features. It requires updating only centers of UBM. The testing is a fast scoring process described in (Reynolds et al., 2000) to score the input utterance to each adapted foreign accent models by selecting top five Gaussians per speech frame.

Table 9 shows the result of GMM-UBM system with four different UBM sizes. Increasing the number of Gaussians results in higher recognition accuracy. Table 10 further compares the best recognition accuracies achieved by both recognizers. In the i-vector system, the best recognition accuracy, i.e. EER_{avg} of 12.60%, is achieved with all the hyper-parameters trained from the FSD corpus and HLDA, WCCN and score normalization being turned on. On the other hand, the best GMM-UBM recognition accuracy, EER_{avg} of 17.00%, is achieved with UBM order 2048 when score normalization is applied. The results indicate that the i-vector system outperforms the conventional GMM-UBM system with 25% relative improvements in terms of EER_{avg} at the cost of higher computational time and additional development data.

4.3. Detection performance per target language

In the previous section, we analyzed the overall average recognition accuracy. Now, here we focus on performance for each individual foreign accent. In order to compensate the lack of sufficient development data in reporting these results, we used the previously unused accents in the FSD corpus to train UBM, T-matrix and HLDA. These unused accents are Chinese, Dari, Finnish, French, Italian, Somali, Swedish and Misc[5] corresponding to 210 speakers and 1110 utterances in total. Further, to increase the number of test trials in the classification stage, we report the results using a leave-one-speaker-out (LOSO) protocol. As demonstrated in the pseudo code below, for every accent, each speaker's utterances are held out one at a time and the remaining utterances are used in modeling the $\hat{\mathbf{w}}_{target}$ as in Eq. (5). The held-out utterances are used as the evaluation utterances.

[5] Refers to those utterances in which the spoken foreign accent is not clear.

Table 8
The joint effect of WCCN and HLDA on the recognition accuracy. (Score normalization turned on.)

HLDA	WCCN	EER_{avg}%	$C_{avg} \times 100$	Id_{error}%
No	No	17.70	7.04	39.58
Yes	No	13.01	6.94	32.85
No	Yes	19.00	7.31	41.55
Yes	Yes	**12.60**	**6.85**	**30.85**

Algorithm 1. Leave-one-speaker-out (LOSO)

Let $A = \{a_1, a_2, \ldots, a_L\}$ be the set of L target accents
Let $S(a_i)$ be the set of speakers in target accent a_i
$\hat{\mathbf{w}}^{a}_{\text{target}}$ defines the i-vectors of target accent a after HLDA and WCCN.
for $a_i \in A$ **do**
  for $s_j \in S(a_i)$ {Held-out test speaker} **do**
    Let $S' = S(a_i) - s_j$ {Remove the speaker being tested}
    Form $\hat{\mathbf{w}}^{a}_{\text{target}}$ using the i-vectors in set S', Eq. (5)
    Compute cosine scores $\langle \mathbf{w}^{s_j}_{\text{test}}, \mathbf{w}^{a}_{\text{target}} \rangle$ {$\mathbf{w}^{s_j}_{\text{test}}$ are the test i-vectors of speaker s_j}
  end for
end for
Normalize scores per each target accent, Eq. (6)

Table 11 shows the language wise results. The results suggest that certain languages which do not belong to the same sub-family as Finnish are easier to detect. Turkish achieves the highest recognition accuracy, whereas English shows highest error rate. The recognition accuracy is consistent among Albanian, Arabic, Kurdish and Russian languages. C_{avg} is bigger than the results already given in Table 10. Note that in Table 11, the unused accents are used to train UBM, T-matrix and HLDA. This induces mismatch between model training data and the hyper-parameter training data. Which is not the case in Table 10.

Fig. 3 further exemplifies the distribution of scores for three selected languages of varying detection difficulties. The histograms are plotted with the same number of bins, 50. For visualization purposes, the width of bins in the non-target score histogram was set smaller than in the target score histogram. The score distribution explains the differences between EERs. For example, in case of Turkish as the easiest and English as the most difficult detected accent,

Table 9
Recognition performance of GMM-UBM system with different UBM sizes.

UBM size	EER_{avg}%	$C_{avg} \times 100$
256	19.94	11.02
512	19.03	10.56
1024	18.20	10.12
2048	**17.00**	**9.46**

Table 10
Comparison between the best recognition accuracy in the GMM-UBM and i-vector system. (Score normalization turned on for the both cases.)

Recognition system	EER_{avg}%	$C_{avg} \times 100$	Id_{error}%
GMM-UBM	17.00	9.46	43.65
i-vector	**12.60**	**6.85**	**30.85**

the overlap between the target and the non-target scores is higher in the latter.

Here, the problem is treated as foreign accent identification task. Table 12 displays the confusion matrix corresponding to Table 11. In all the cases, majority of the detected cases corresponds to the correct class (i.e., the entries in the diagonal). Taking Turkish as the language with the highest recognition accuracy, out of the 11 misclassified Turkish test segments, 7 were misclassified as Arabic. This might be because Turkey is bordered by two Arabic countries, Syria and Iraq, and Turkish shares common features with Arabic. Regarding Spanish, out of the 27 misclassified test segments, 9 were detected as Arabic. It is possibly due to the major influence of Arabic on Spanish. In particular, numerous words of Arabic origin are adopted in the Spanish language.

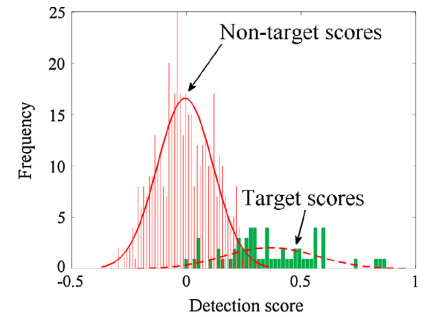
To analyze further reasons why some languages are harder to detect, we first compute the average target language score on a speaker-by-speaker basis. To measure the degree of speaker variation, we show the standard deviation of these average scores in Table 13, along with the corresponding EER and C_{DET} values. The results indicate that languages with more diverse speaker populations, having speaker-dependent biases in the detection scores, are more difficult to handle. It does not yet explain why certain languages, such as Russian, have a larger degree of speaker variation, but suggests that there will be space for further research in speaker normalization techniques.

4.4. Factors affect foreign accent recognition

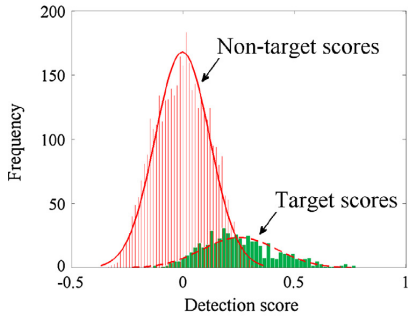
We are interested to find out what factors affect the foreign accent recognition accuracies. The rich metadata available in the FSD corpus includes language proficiency, speaker's age, education and the place where the second language is spoken. In the following analysis, we used the

Table 11
Per language results in terms of EER% and $C_{DET} \times 100$ for the i-vector system.

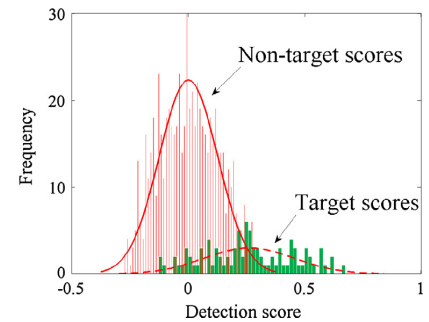
Accents	EER%	$C_{DET} \times 100$
Turkish	11.90	6.35
Spanish	16.49	6.92
Albanian	18.76	7.00
Arabic	18.98	7.17
Kurdish	19.37	7.19
Russian	19.68	7.21
Estonian	20.05	7.52
English	23.60	8.00



(a) Turkish



(b) Russian



(c) English

Fig. 3. Distribution of scores for Turkish, Russian and English accents.

whole set of scores from the LOSO experiment and grouped them to different categories according to each metadata variable at a time.

Language proficiency

To find out the impact of language proficiency, we take the sum of spoken and written Finnish grades in the FSD corpus as a proxy of the speaker's Finnish language proficiency. The objective was to find out how speakers' language proficiency and their detected foreign accent are related. Fig. 4 shows C_{avg} for each grade group. As hypothesized, the lowest C_{avg} is attributed to speakers with the lower grade (5) and the highest accuracy to speakers with the higher grade (8). This indicates that detecting the foreign accents from speakers with higher proficiency in Finnish is considerably more difficult than speakers with lower proficiency.

In addition, we looked at language proficiency across different target languages. We study the average language proficiency grade across the speakers in different languages (Table 14). For the three most difficult languages to detect, Russian, Estonian and English, the average language proficiency grades are higher than the rest of languages, supporting the preceding analysis.

Age of entry

Age is one of the most important effective factors in learning a second language (Krishna, 2008). The common notion is that younger adults learn the second language more easily than older adults. (Larsen-Freeman, 1986) argues that during the period of time between birth and the age when a children enters puberty, learning a second language is quick and efficient. In the second language acquisition process, one of the affecting factors relates to the experience of immigrants, such as the age of entry and the length of residence (Krishna, 2008). We analyze the relationship between the age of entry and the foreign accent recognition results. To analyze the effect of age to foreign accent detection, we categorized the detection scores into six age groups with 10 years age interval (Fig. 5). Our hypothesis was that mother tongue detection is easier in older people than younger ones. The results support this hypothesis. C_{avg} decreases from 5.30 (a relative

Table 12
Confusion matrix of the results corresponding to Table 11.

	Predicted label							
	Turk.	Span.	Alba.	Arab.	Kurd.	Russ.	Esto.	Engl.
True label								
Turk.	50	0	1	7	0	1	0	2
Span.	1	58	1	11	2	3	7	2
Alba.	1	0	61	9	1	5	11	1
Arab.	4	2	14	110	7	7	12	4
Kurd.	5	1	1	5	50	6	3	6
Russ.	51	21	51	26	2	369	13	28
Esto.	5	5	7	15	1	6	117	15
Engl.	7	3	3	6	3	7	9	59

Table 13
The standard deviation of the average target language score on a speaker-by-speaker basis along with the corresponding EER and C_{DET} results.

Accents	Standard deviation	EER%	$C_{DET} \times 100$
Turkish	0.1205	11.90	6.35
Spanish	0.1369	16.49	6.92
Albanian	0.1380	18.76	7.00
Arabic	0.1505	18.98	7.17
Kurdish	0.1392	19.37	7.19
Russian	0.1402	19.68	7.21
Estonian	0.1621	20.05	7.52
English	0.1667	23.60	8.00

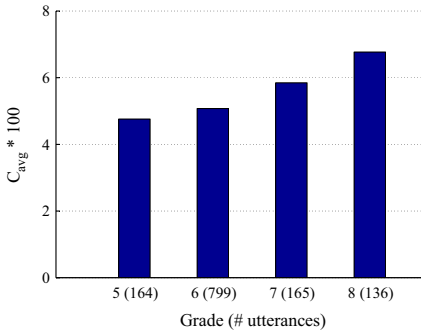


Fig. 4. $C_{avg} \times 100$ for different grade groups in the language proficiency measurement.

Table 14
The average language proficiency grade across the speakers in different languages along with the corresponding EER and C_{DET} results.

Accents	Grade	EER%	$C_{DET} \times 100$
Turkish	6.09	11.90	6.35
Spanish	6.20	16.49	6.92
Albanian	5.78	18.76	7.00
Arabic	5.73	18.98	7.17
Kurdish	5.71	19.37	7.19
Russian	6.30	19.68	7.21
Estonian	7.02	20.05	7.52
English	6.34	23.60	8.00

decrease of 16%) to 4.45 from the age group [11–20] to [61–70]. This indicates that the mother tongue detection in older age groups could be easier than in the younger age groups.

Level of education

According to Gardner's socio-educational model (Gardner, 2010), intrinsic motivation to learn a second language is strongly correlated to educational achievements. The objective was to find out how speakers' level of education and their detected foreign accent might be related. To analyze the effect of education, we categorized the detection scores into different levels of education groups. We hypothesized that people with higher level of education

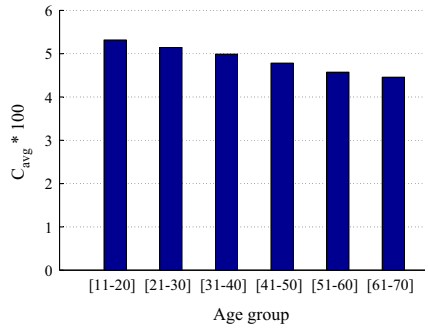


Fig. 5. $C_{avg} \times 100$ for different age groups. Age refers to age of entry to foreign country. Number of utterances for the age group [11–20], [21,30], ..., [61–70] is 46, 342, 535, 239, 100, 12, respectively.

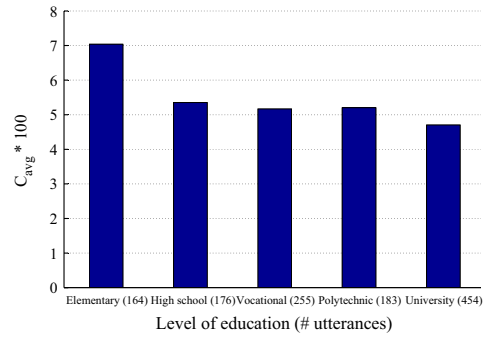


Fig. 6. $C_{avg} \times 100$ for different level of education groups.

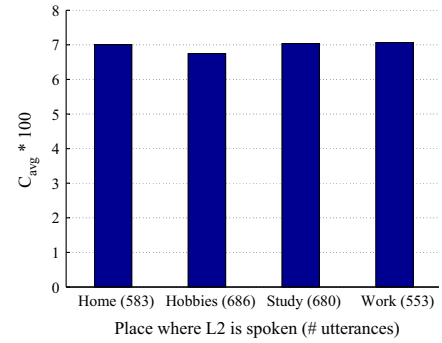


Fig. 7. $C_{avg} \times 100$ for different places where the second language is spoken.

speak the second language more fluently than lower educated people. As a consequence, mother tongue detection for higher educated people is relatively difficult. But the results in Fig. 6 in fact show the opposite; the highest C_{avg} belongs to elementary school and the lowest to university education. However, C_{avg} is somewhat similar for the high school, vocational school, and polytechnic level of education.

Where second language is spoken

Finally, we were also interested to observe whether the place or situation, where the second language is spoken, affects foreign accent detection or not. To this end, we categorized the scores into four groups based on the level of social interaction: home, hobbies, study and work. We hypothesized that the places with more social interactions between people, the mother tongue traits will be less in the second spoken language, therefore making it more difficult to detect the mother tongue. Fig. 7 shows the C_{avg} for different places where the second language is spoken. The results indicate no considerable sensitivity to the situation where the second language is spoken.

5. Conclusion

In this work, we studied how the various i-vector extractor parameters, data set selections and the speaker's language proficiency affects foreign accent detection accuracy. Regarding **parameters**, highest accuracy was achieved using UBMs with 512 Gaussians, i-vector dimensionality of 1000 and HLDA dimensionality of 180. These are similar to those reported in general speaker and language recognition literature, except for the higher-than-usual i-vector dimensionality of 1000.

Regarding **data**, we found that the choice of the UBM training data is the most critical part, followed by T-matrix and HLDA. This is understandable since the earlier system components affect the quality of the remaining steps. In all cases, the error rates increased unacceptably high for mismatched sets of hyper-parameter training. Thus, our answer to the question whether hyper-parameters could be reasonably trained from mismatched language and channel is negative. The practical implication of this is that the i-vector approach, even though producing reasonable accuracy, requires careful data selection for hyper-parameter training – and this is not always feasible.

Applying within-class covariance normalization followed by score normalization technique further increased the i-vector system performance by 6% relative improvements in terms of C_{avg}. We also showed that the i-vector system outperforms the conventional GMM-UBM system by 28% relative decrease in terms of C_{avg}.

In our view, the most interesting contribution of this work is the analysis of **language aspects**. The results, broken down by the accents, clearly suggested that certain languages which do not belong to the same sub-family as Finnish are easier to detect. Turkish was the easiest (C_{DET} of 6.35) while for instance Estonian, a language similar to Finnish, yielded C_{DET} of 7.52. The most difficult language was English with C_{DET} of 8.00. In general, confusion matrix revealed that phonetically similar languages are more often confused.

Our analysis on affecting factors suggested that language proficiency and age of entry affect detection performance. Specifically, accents produced by fluent speakers of Finnish are more difficult to detect. Speaker group with the lowest language grade 5 yielded C_{avg} of 4.75 while the group with grade 8 yielded C_{avg} of 6.76. Analysis of the age of entry, in turn, indicated that mother tongue detection in older speakers is easier than younger speakers. The age group [61–70] years yielded C_{avg} of 4.45 while the group with age interval [11–20] years old yielded C_{avg} of 5.31.

After optimizing all the parameters, the overall EER_{avg} and C_{avg} were 12.60% and 6.85, respectively. These are roughly an order of magnitude higher compared to state-of-the-art text-independent speaker recognition with i-vectors. This reflects the general difficulty of the foreign accent detection task, leaving a lot of space for future work on new feature extraction and modeling strategies. While these values are unacceptably high for security applications, the observed correlation between language proficiency and recognition scores suggests potential applications for automatic spoken language proficiency grading.

Acknowledgements

We would like to thank Ari Maijanen from University of Jyväskylä for an immense help with the FSD corpus. This work was partly supported by Academy of Finland (projects 253000, 253120 and 283256) and Kone Foundation – Finland.

References

Bahari, M.H., Saeidi, R., hamme, H.V., Leeuwen, D.V., 2013. Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, May 26–31. Vancouver, BC, Canada, pp. 7344–7348.

Behravan, H., 2012. Dialect and Accent Recognition. Master's Thesis, School of Computing, University of Eastern Finland, Joensuu, Finland.

Behravan, H., Hautamäki, V., Kinnunen, T., 2013. Foreign accent detection from spoken Finnish using i-Vectors. In: INTERSPEECH 2013: 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25–29, pp. 79–83.

Brümmer, N., van Leeuwen, D., 2006. On calibration of language recognition scores. In: IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, June 28–30, pp. 1–8.

Burget, L., Matejka, P., Schwarz, P., Glembek, O., Cernocký, J., 2007. Analysis of feature extraction and channel compensation in a GMM speaker recognition system. IEEE Trans. Audio, Speech Lang. Process. 15 (7), 1979–1986.

Canavan, A., Zipperle, G., 1996. CallFriend Corpus. <http://yki-korpus.jyu.fi/> (Accessed 04.07.13).

Chen, N.F., Shen, W., Campbell, J.P., 2010. A linguistically-informative approach to dialect recognition using dialect-discriminating context-dependent phonetic models. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Sheraton Dallas Hotel, Dallas, Texas, USA, March 14–19, pp. 5014–5017.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011a. Front-end factor analysis for speaker verification. IEEE Trans. Audio, Speech Lang. Process. 19 (4), 788–798.

Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D.A., Dehak, R., 2011b. Language recognition via i-vectors and dimensionality reduction. In: INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, pp. 857–860.

DeMarco, A., Cox, S.J., 2012. Iterative classification of regional British accents in I-vector space. In: Machine Learning in Speech and Language Processing (MLSLP), Portland, OR, USA, September 14–18, pp. 1–4.

Flege, J.E., Schirru, C., MacKay, I.R.A., 2003. Interaction between the native and second language phonetic subsystems. Speech Commun. 40 (4), 467–491.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, second ed. Academic Press.

Gales, M.J.F., 1999. Semi-tied covariance matrices for hidden Markov models. IEEE Trans. Speech Audio Process. 7 (3), 272–281.

GAO, 2007. Border Security: Fraud Risks Complicate States Ability to Manage Diversity Visa Program. DIANE Publishing.

Garcia-Romero, D., Espy-Wilson, C.Y., 2011. Analysis of i-vector length normalization in speaker recognition systems. In: INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, pp. 249–252.

Gardner, R.C., 2010. Motivation and Second Language Acquisition: The Socio-educational Model. Peter Lang, New York.

González, D.M., Plchot, O., Burget, L., Glembek, O., Matejka, P., 2011. Language recognition in ivectors space. In: INTERSPEECH 2011: 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, pp. 861–864.

Grosjean, F., 2010. Bilingual: Life and Reality. Harvard University Press.

Hatch, A.O., Kajarekar, S.S., Stolcke, A., 2006. Within-class covariance normalization for SVM-based speaker recognition. In: INTERSPEECH 2006, ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17–21, pp. 1471–1474.

Hermansky, H., Morgan, N., 1994. RASTA processing of speech. IEEE Trans. Speech Audio Process. 2 (4), 578–589.

Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., Mason, M., 2011. i-vector based speaker recognition on short utterances. In: INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, pp. 2341–2344.

Kenny, P., 2005. Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms. Technical Report CRIM-06/08-13.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P., 2008. A study of interspeaker variability in speaker verification. IEEE Trans. Audio, Speech Lang. Process. 16 (5), 980–988.

Kohler, M.A., Kennedy, M., 2002. Language identification using shifted delta cepstra. In: 45th Midwest Symposium on Circuits and Systems, vol. 3, pp. III-69–72.

Krishna, B., 2008. Age as an affective factor in second language acquisition. Engl. Specif. Purp. World 21 (5), 1–14.

Kumar, N., 1997. Investigation of Silicon-auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition. Ph.D. Thesis, Baltimore, Maryland.

Kumpf, K., King, R.W., 1997. Foreign speaker accent classification using phoneme-dependent accent discrimination models and comparisons with human perception benchmarks. In: Fifth European Conference on Speech Communication and Technology, EUROSPEECH, Rhodes, Greece, September 22–25, pp. 2323–2326.

Larsen-Freeman, D., 1986. Techniques and Principles in Language Teaching. Oxford University Press, New York.

Lee, L., Rose, R.C., 1996. Speaker normalization using efficient frequency warping procedures. In: Proceedings of the Acoustics, Speech, and Signal Processing, May 7–10, pp. 353–356.

Li, H., Ma, B., Lee, K.-A., 2013. Spoken language recognition: from fundamentals to practice. Proc. IEEE 101 (5), 1136–1159.

Loog, M., Duin, R.P.W., 2004. Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion. IEEE Trans. Pattern Anal. Mach. Intell. 26 (6), 732–739.

Martin, A.F., Doddington, G.R., Kamm, T., Ordowski, M., Przybocki, M.A., 1997. The DET curve in assessment of detection task performance. In: EUROSPEECH 1997, 5th European Conference on Speech Communication and Technology, Rhodes, Greece, September 22–25, pp. 1895–1898.

Munoz, C., 2010. On how age affects foreign language learning. Adv. Res. Lang. Acquisit. Teach., 39–49.

Rao, W., Mak, M.-W., 2012. Alleviating the small sample-size problem in i-vector based speaker verification. In: 8th International Symposium on Chinese Spoken Language Processing, Kowloon Tong, China, December 5–8, pp. 335–339.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted Gaussian mixture models. Digital Signal Process. 10 (1–3), 19–41.

Rouvier, M., Dufour, R., Linarès, G., Estève, Y., 2010. A language-identification inspired method for spontaneous speech detection. In: INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Japan, September 26–30, pp. 1149–1152.

Scharenborg, O., Witteman, M.J., Weber, A., 2012. Computational modelling of the recognition of foreign-accented speech. In: INTERSPEECH 2012: 13th Annual Conference of the International Speech Communication Association, September 9–13, pp. 882–885.

Torres-Carrasquillo, P.A., Gleason, T.P., Reynolds, D.A., 2004. Dialect identification using Gaussian mixture models. In: Proceeding Odyssey: The Speaker and Language Recognition Workshop, May 31–June 3, pp. 757–760.

University of Jyväskylä, 2000. Finnish National Foreign Language Certificate Corpus, University of Jyväskylä, Centre for Applied Language Studies. <http://yki-korpus.jyu.fi/>.

Witteman, M., 2013. Lexical Processing of Foreign-accented Speech: Rapid and Flexible Adaptation. Ph.D. Thesis.

Wu, T., Duchateau, J., Martens, J., Compernolle, D., 2010. Feature subset selection for improved native accent identification. Speech Commun. 52 (2), 83–98.

Zissman, M.A., 1996. Comparison of four approaches to automatic language identification of telephone speech. IEEE Trans. Speech Audio Process. 4 (1), 31–44.

Paper **II**

H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C.-H. Lee
"Introducing Attribute Features to Foreign Accent Recognition"
in Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP),
pp. 5332–5336, Florence, Italy, 2014.

INTRODUCING ATTRIBUTE FEATURES TO FOREIGN ACCENT RECOGNITION

Hamid Behravan[1], *Ville Hautamäki*[1], *Sabato Marco Siniscalchi*[2,3], *Tomi Kinnunen*[1] *and Chin-Hui Lee*[3]

[1]School of Computing, University of Eastern Finland, Finland
[2]Faculty of Architecture and Engineering, University of Enna "Kore", Italy
[3]School of ECE, Georgia Institute of Technology, USA

ABSTRACT

We propose a hybrid approach to foreign accent recognition combining both phonotactic and spectral based systems by treating the problem as a spoken language recognition task. We extract speech attribute features that represent speech and acoustic cues reflecting foreign accents of a speaker to obtain feature streams that are modeled with the i-vector methodology. Testing on the Finnish Language Proficiency exam corpus, we find our proposed technique to achieve a significant performance improvement over the state-of-the-art systems using only spectral based features.

Index Terms— Speech attributes, i-vector, foreign accent recognition, language recognition

1. INTRODUCTION

In *automatic foreign accent recognition*, we aim to detect speaker's mother tongue (L1) when he or she is speaking in another language (L2) [1]. When speaking in L2, the speaker's accent is usually colored by the learned patterns in L1 [2]. When the native language is spoken instead, it can be said to vary in terms of its regional dialects and accents. *Dialect* refers to linguistic variations of a language, while *accent* refers to different ways of pronouncing a language within a community [3]. In the NIST *language recognition evaluation* (LRE) scenarios, dialect and accent recognition have been included as sub-tasks. As an example, the most recent LRE 2011 covered four different Arabic dialects as target languages [4]. Foreign accent recognition, however, differs from common accent recognition in two major distinctions. Firstly, non-native speaker's *accentedness* partly depends on the language proficiency [2]. Secondly, the L2 is a noisy channel through which the identity of the mother tongue is transmitted.

In this study we treat foreign accent recognition as a language recognition task typically accomplished via either *acoustic* or *phonotactic* modeling [5]. In the former approach, acoustic features, such as *shifted delta cepstra* (SDC), are used with bag-of-frames models, such as *universal background model* (UBM) with adaptation [6, 7]. The latter is based on the hypothesis that dialects or accents differ in terms of their phone sequence distributions. It uses phone recognizer outputs, such as n-gram statistics, together with a language modeling back-end [8, 9].

Among the choices for acoustic modeling, the recent *i-vector* paradigm [10] has proven successful in both speaker [10, 11], language [12], and accent recogntion [13]. It extracts a low-dimensional representation of the sequence of feature vectors. Session and channel variability is typically tackled with techniques such as *linear discriminant analysis* (LDA). The i-vectors from spectral features have been used in dialect and foreign accent characterization. In [14], L1 of the non-native English speakers was recognized using multiple spectral systems, including i-vectors with different back-ends. The i-vector based system outperformed other compared methods most of the time. In [1], it was found out that the i-vector system using SDCs outperformed other methods in recognizing Finnish non-native accents.

In language recognition, spectral features with i-vector based systems have been seen to outperform the classical phonotactic language recognition [4]. However, *knowledge based* modeling, such as phonotactic features, are known to be linguistically and phonetically relevant [5]. However, the front-end of the phonotactic system needs a tokenizer that will turn the utterance into a sequence of "phonetic letters" [15, 16]. An ad-hoc approach is to use a phone recognizer developed for one language, such as Hungarian, and apply it to all phonotactic recognition tasks [17].

In the present work, we argue that, especially in foreign accent recognition, a universal phonetic tokenizer is preferable. It will be able to find differences between the unknown L1 and the known L2. For example, Spanish L1 speaker trying to pronounce Finnish word "*stressi*" (stress) will typically lead to /e/ placed as a prefix, leading to "*estressi*". In this case, detecting a vowel in the beginning of the word is a cue for Spanish L1. We then propose to use speech attributes [18, 19, 20] to represent a language-universal set of units to be modeled. In addition, we avoid the early quantization of the attribute detector scores by computing an i-vector from the detector score vector streams.

This work was partially supported by Academy of Finland (projects 253000 and 253120).

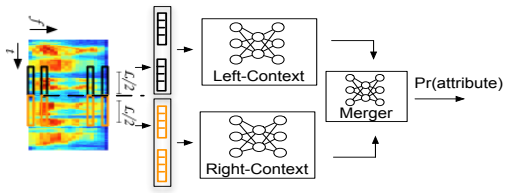


Fig. 1. The internal structure of an attribute detector is shown. Energy trajectories are fed into the left-context and right-context ANNs. A merger then combines the outputs generated by those two neural networks and produced the final attribute posterior probabilities.

2. SPEECH ATTRIBUTE EXTRACTION

2.1. Choice and Extraction of Attribute Features

The set of speech attributes used in this work is mainly acoustic phonetic features, and it comprises five manner of articulation classes (**glide, fricative, nasal, stop, and vowel**), and **voicing**. Those attributes could be identified from a particular language and shared across many different languages, so they could also be used to derive a universal set of speech units. Furthermore, data-sharing across languages at the acoustic phonetic attribute level is naturally facilitated by using these attributes, so more reliable language-independent acoustic parameter estimation can be anticipated [21]. In [16], it was also shown that these attributes can be used to compactly characterize any spoken language along the same lines as in the automatic speech attribute transcription (ASAT) paradigm for automatic speech recognition (ASR) [20]. Therefore, we believe that it can also be useful to characterize speaker accent.

Data-driven detectors are used to spot speech cues embedded in the speech signal. An attribute detector converts an input utterance into a time series that describes the level of presence (or level of activity) of a particular property of an attribute over time. A bank of six detectors is used in this work, each detector is individually designed for spotting of a particular event. Each detector is realized with three single hidden layer feed-forward ANNs (artificial neural networks) organized in a hierarchical structure and trained on sub-band energy trajectories that are extracted with a 15 band uniform mel-frequency filterbank. For each critical band a window of $310ms$ centered around the frame being processed is considered and split in two halves: left-context and right-context [22]. Two independent front-end ANNs ("lower nets") are trained on those two halves and generate left- and right-context speech attribute posterior probabilities, respectively. The outputs of the two lower nets are then sent to the third ANN that acts as a merger and gives the attribute-state posterior probability of the target speech attribute. Figure 1 shows the detector architecture in detail.

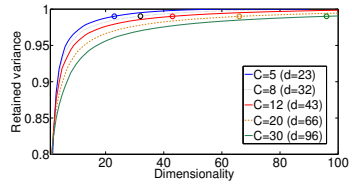


Fig. 2. Remaining variance after PCA. Comparing stacked context sizes 5, 8, 12, 20 and 30 frames.

2.2. Long-term Attribute Extraction

Each attribute detector outputs probabilities $p(H^{(i)}_{\text{target}}|\boldsymbol{f})$, $p(H^{(i)}_{\text{anti}}|\boldsymbol{f})$ and $p(H^{(i)}_{\text{noise}}|\boldsymbol{f})$, of target class i, non-target and noise model, given a speech frame $\boldsymbol{f}$. All these probabilities sum to one. We then form a new feature vector $\boldsymbol{x}$ by concatenating each of these posteriors of each six target classes. Since language and dialect recognizers benefit from an inclusion of long temporal context, it is natural to study similar ideas for attribute modeling. The first idea is to compute SDCs from the attribute features, treating them analogous to cepstral coefficients. But since this is difficult to interpret, we study a simple feature stacking. To this end, let $\boldsymbol{x}(t)$ denote the 18-dimensional (6 attributes $\times$ 3 features) attribute vector at frame t. We form a sequence of new $p = 18 \times C$ dimensional stacked vectors $\tilde{\boldsymbol{x}}_C(t) = (\boldsymbol{x}(t)^*, \boldsymbol{x}(t+1)^*, \ldots, \boldsymbol{x}(t+C-1)^*)^*,\ t = 1, 2, \ldots,$ where C is the context size and * stands for transpose. Principal component analysis (PCA) is used to project each $\tilde{\boldsymbol{x}}_C(t)$ onto the first $d \ll p$ eigenvectors corresponding to the largest eigenvalues of the sample covariance matrix. We estimate the PCA basis from the same data as the UBM and the T-matrix, after VAD.We set d to retain 99 % of the cumulative variance. As Fig. 2 indicates, d varies from $\sim$20 to $\sim$100, with larger dimensionality assigned to longer context as one expects.

3. RECOGNIZING FOREIGN ACCENTS

3.1. I-vector Modeling

We now shortly review i-vector extraction. It is grounded on the *universal background model* (UBM), which is a M-component Gaussian mixture model parametrized by $\{w_m, \boldsymbol{m}_m, \boldsymbol{\Sigma}_m\}, m = 1, \ldots, M$, where we have mixture weight, mean vector and covariance matrix, respectively. We here restrict the covariance matrix to be diagonal. The i-vector model is defined for the UBM component m as [10]:

$$\boldsymbol{s}_m = \boldsymbol{m}_m + \boldsymbol{V}_m \boldsymbol{y} + \epsilon_m, \tag{1}$$

where $\boldsymbol{V}_m$ is the sub-matrix of the total variability matrix, $\boldsymbol{y}$ is the latent vector, called an i-vector, ϵ_m is the residual term and $\boldsymbol{s}_m$ is the m'th sub-vector of the utterance dependent supervector. The ϵ_m is distributed as $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_m)$, where

$\boldsymbol{\Sigma}_m$ is a diagonal matrix. Given all these definitions, posterior density of the $\boldsymbol{y}$, given the sequence of observed feature vectors, is Gaussian. Expectation of the posterior is the extracted i-vector. Hyperparameters of the i-vector model, $\boldsymbol{m}_m$ and $\boldsymbol{\Sigma}_m$ are copied directly from UBM and $\boldsymbol{V}_m$ are estimated by EM algorithm from the same corpus as is used to estimate the UBM.

3.2. Scoring against Accent Models

We use *cosine scoring* [23] between two i-vectors $\boldsymbol{y}_{\text{test}}$ and $\boldsymbol{y}_{\text{target}}$ to match test utterance to target L2 language model. Cosine score is given by the dot product $\langle \hat{\boldsymbol{y}}_{\text{test}}, \hat{\boldsymbol{y}}_{\text{target}} \rangle$,

$$\text{score}(\boldsymbol{y}_{\text{test}}, \boldsymbol{y}_{\text{target}}) = \frac{\hat{\boldsymbol{y}}_{\text{test}}^{\text{T}} \cdot \hat{\boldsymbol{y}}_{\text{target}}}{\|\hat{\boldsymbol{y}}_{\text{test}}\| \, \|\hat{\boldsymbol{y}}_{\text{target}}\|}, \tag{2}$$

where $\boldsymbol{A}$ is the HLDA projection matrix trained by using all training utterances and $\hat{\boldsymbol{y}}_{\text{test}}$ is,

$$\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{A}^{\text{T}} \boldsymbol{y}_{\text{test}}. \tag{3}$$

In order to model $\hat{\boldsymbol{y}}_{\text{target}}$, we followed the same strategy used in [4], where $\hat{\boldsymbol{y}}_{\text{target}}$ is defined as

$$\hat{\boldsymbol{y}}_{\text{target}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \hat{\boldsymbol{y}}_{id}, \tag{4}$$

where N_d is the number of training utterances in dialect d, and $\hat{\boldsymbol{w}}_i$ is the projected i-vector of training utterance i for accent d computed the same way as in (3).

4. EXPERIMENTAL SETUP

4.1. Corpora

The "stories" part of the OGI Multi-language telephone speech corpus [24] was used to train the articulatory detectors. This corpus has phonetic transcriptions for six languages: English, German, Hindi, Japanese, Mandarin, and Spanish. Data from each language were pooled together to obtain: 5.57 hours for the training set, and 0.52 hours for the validation set.

A series foreign accent recognition experiments was performed on the *FSD* corpus [25] which was developed to assess Finnish language proficiency among adults of different nationalities. These selected the oral responses portion of the exam, corresponding to 18 foreign accents. Since the number of utterances is small, 9 accents — Russian, Albanian, Arabic, Chinese, English, Estonian, Kurdish, Spanish, and Turkish — with enough available data were used. The unused accents are, however, used in training the UBM and the V_m-matrices. For our purposes, each accent set is randomly split into a test and a train set. The test set consists of (approximately) 30% of the utterances, while the training set consists of the remaining 70% to train foreign accent recognizers. The raw audio files were partitioned into 30 sec chunks and re-sampled to 8 KHz. Statistics of the test and train portions are shown in Table 1.

Table 1. Train and test file distributions in the FSD corpus.

Accent	#train files	#test files	#speakers
Spanish	60	25	15
Albanian	67	30	19
Kurdish	83	35	21
Turkish	84	34	22
English	92	37	23
Estonian	153	63	38
Arabic	166	67	42
Russian	599	211	235

Table 2. Sliding window context experiments with PCA as a dimensionality reduction.

PCA features	Pooled EER (%)	$C_{\text{avg}} \times 100$
$(C = 5, d = 23)$	10.65	4.82
$(C = 20, d = 50)$	10.44	4.71
$(C = 30, d = 96)$	**8.73**	**4.47**

4.2. Attribute Detector Design

One-hidden-layer feed forward multi-layer perceptrons (MLPs) were used to implement each attribute detector shown in Figure 1. The number of hidden nodes with a sigmoidal activation function is 500. MLPs were trained to estimate attribute posteriors, and the training data were separated into "feature present," "feature absent," and "other" regions for every phonetic class used in this work. The classical back-propagation algorithm with a cross-entropy cost function was adopted to estimates the MLP parameters. To avoid over-fitting, the reduction in classification error on the development set was adopted as the stopping criterion. The attribute detectors employed in this work were actually just those used in [21].

4.3. Evaluation Protocol

System performance is reported in terms of *equal error rate* (EER) and average detection cost (C_{avg}) [5]. Results are reported per each accent for a cosine scoring classifier. C_{avg} is defined as [5],

$$C_{\text{avg}} = \frac{1}{J} \sum_{j=1}^{M} C_{\text{DET}}(L_j), \tag{5}$$

where $C_{\text{DET}}(L_j)$ is the detection cost for subset of test segments trials for which the target accent is L_j and J is the

Table 3. Summary of results and compared against baseline spectral system, results are shown in pooled EER and C_{avg}.

Features (dimensionality)	Pooled EER (%)	$C_{\text{avg}} \times 100$
SDC+MFCC(56)	15.00	7.00
Attribute(18)	12.54	5.07
Attribute+Δ(36)	11.33	4.79
Attribute+Δ+$\Delta\Delta$(54)	11.00	4.59
PCA features(96)	**8.73**	**4.47**

number of target languages. The per target accent cost is then,

$$C_{\text{DET}}(L_j) = C_{\text{miss}} P_{\text{tar}} P_{\text{miss}}(L_j) + C_{\text{fa}}(1 - P_{\text{tar}}) \frac{1}{J-1} \sum_{k \neq j} P_{\text{fa}}(L_j, L_k). \quad (6)$$

The miss probability (or false rejection rate) is denoted by P_{miss}, i.e., a test segment of accent L_i is rejected as being in that accent. On the other hand $P_{\text{fa}}(L_i, L_k)$ denotes the probability when a test segment of accent L_k is accepted as being in accent L_i. It is computed for each target/non-target accent pairs. Measures, C_{miss} and C_{fa}, are costs of making errors and both were set to 1. P_{tar} is the prior probability of a target accent and was set to 0.5.

5. EXPERIMENTS AND RESULTS

First we experimented with different context sizes ($C = 5, 20, 30$). Feature vectors were concatenated and PCA dimensionality reduction was trained on the held out data. Output dimensionality (d) was set to retain 99% percent of the cumulative variance. In Table 2 we see that increasing the context size from 5 to 30 will decrease the both pooled EER and C_{avg}. We also attempted to use context as large as 40 frames, which resulted to a numerical problems in UBM computation. Output dimensionality of 124 was too large with respect to the available data, so we observed singular Gaussian components.

We applied the context size 30 to the following experiments (see Table 3). We contrasted the above mentioned system to the baseline SDC+MFCC based system in [1]. In addition to sliding window based context modeling, we also employ standard Δ and $\Delta\Delta$ to attribute feature vectors. We notice that increasing the context size using Δ and $\Delta\Delta$ features improves marginally over not using the context at all. A large 30-frame context brought forth an improvement. All systems based on speech attributes improved substantially over the baseline. In Table 4 we show the per target accent error rates, in EER and C_{DET}. We notice that there is a large variation in error rates, where Turkish and Albanian are easiest and Russian and Estonian are the hardest to recognize.

We also studied the relative importance of individual speech attributes to system performance in Fig. 3. No context was used, so raw pooled EER is 12.54%. We left out one by one all attributes, so we had 15-dimensional feature vectors. We noticed that voicing, stop and vowels are individually beneficial (leaving any one of them out will decrease the system performance). On the other hand, glide, nasal and fricative are not individually useful. We also noticed that in terms of conclusions, pooled EER and C_{avg} agree. Usefulness of vowels in contrast to other features can be explained by the fact that Finnish has a very large vowel space (with 8 vowels) including vowel lengthening. It can create difficulties for L2 speakers to hit the correct vowel target, thus showing the L1 influence.

Table 4. Per-language results for PCA features (30,96). The results are given in EER and C_{DET}.

Features	EER (%)	$C_{\text{DET}} \times 100$
Spanish	9.00	4.10
Turkish	3.82	2.01
Albanian	4.34	2.48
English	8.11	4.20
Arabic	7.46	4.04
Russian	15.54	8.17
Kurdish	8.57	4.67
Estonian	12.70	6.11

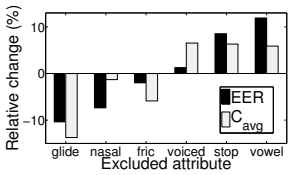


Fig. 3. Exclusion experiment, where relative change is shown when one attribute is left out.

6. CONCLUSION

We proposed speech attributes as features for foreign accent recognition. Instead of using speech attributes directly in a phonotactic system, we modeled the sequence of speech attribute feature vectors using the i-vector methodology. The key idea is to treat foreign accent recognition as a language recognition task and use universal speech attributes. Speech attributes are employed because their statistics can differ considerably from one language to another. Indeed, all attribute feature configurations improved over the spectral-only baseline system. Moreover, adding context information allowed substantially better results. So far, we have only used manner of articulation features, yet place of articulation can further enhance accent recognition performance, as shown in [16]. As a future work, experiments on English foreign accent recognition will be carried out. Furthermore, the possible beneficial effect of combining SDC- and attribute-based information will be investigated.

7. REFERENCES

[1] H. Behravan, V. Hautamäki, and T. Kinnunen, "Foreign accent detection from spoken finnish using i-vectors," in *Interspeech*, Lyon, France, August 2013.

[2] J. Flege, C. Schirru, and I. MacKay, "Interaction between the native and second language phonetic subsystems," *Speech Communication*, vol. 40, no. 4, pp. 467–491, 2003.

[3] J. Nerbonne, "Linguistic variation in computation," in *EACL*, Budabest, Hungary, 2003, pp. 3–10.

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

[5] H. Li, K. A. Lee, and B. Ma, "Spoken language recognition: From fundamentals to practice," *Proceedigns of the IEEE*, vol. 101, no. 5, pp. 1136–1159, May 2013.

[6] P. Torres-Carrasquillo, T. Gleason, and D. Reynolds, "Dialect identification using Gaussian mixture models," in *Speaker Odyssey*, Toledo, Spain, 2004, pp. 757–760.

[7] G. Liu and J. H. Hansen, "A systematic strategy for robust automatic dialect identification," in *EUSIPCO*, Barcelona, Spain, 2011, pp. 2138–2141.

[8] M.A. Zissman, T.P. Gleason, D.M. Rekart, and B.L. Losiewicz, "Automatic dialect identification of extemporaneous conversational latin american spanish speech," in *ICASSP*, Detroit, USA, 1995.

[9] T. Wu, J. Duchateau, J. Martens, and D. Compernolle, "Feature subset selection for improved native accent identification," *Speech Communication*, pp. 83–98, 2010.

[10] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 788–798, 2011.

[11] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances," in *Interspeech*, Florence, Italy, 2011, pp. 2341–2344.

[12] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in i-vectors space," in *Interspeech*, Florence, Italy, 2011, pp. 861–864.

[13] A. DeMarco and S. J. Cox, "Iterative classification of regional British accents in i-vector space," in *Proc. Machine Learningin Speech and Language Processing*, Portland, USA, 2012.

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

[15] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Interspeech*, Brighton, UK, 2009, pp. 168–171.

[16] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Universal attribute characterization of spoken languages for automatic spoken language recognition," *Computer Speech & Language*, vol. 27, no. 1, pp. 209–227, 2013.

[17] K. A. Lee, C. H. You, V. Hautamäki, A. Larcher, and H. Li, "Spoken language recognition in the latent topic simplex," in *Interspeech*, Florence, Italy, 2011, pp. 2893–2896.

[18] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *Interspeech*, Jeju Island, Korea, 2004, pp. 109–112.

[19] I. Bromberg, Q. Fu, J. Hou, J. Li, C. Ma, B. Matthews, A. Moreno-Daniel, J. Morris, S. M. Siniscalchi, Y. Tsao, and Y. Wang, "Detection-based ASR in the automatic speech attribute transcription project," in *Interspeech*, Antwerp, Belgium, 2007, pp. 1829–1832.

[20] C.-H. Lee and S. M. Siniscalchi, "An information-extraction approach to speech processing: Analysis, detection, verification, and recognition," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1089–1115, 2013.

[21] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target specific training data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 875–887, 2012.

[22] P. Schwarz, P. Matějaka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *ICASSP*, Toulouse, France, 2006, pp. 325–328.

[23] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verication," in *Interspeech*, Brighton, UK, 2009, pp. 1559–1562.

[24] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika, "The OGI multi-language telephone speech corpus," in *ICSLP*, Banff, Canada, Oct. 1992, pp. 895–898.

[25] "Finnish national foreign language certificate corpus," http://yki-korpus.jyu.fi.

Paper III

H. Behravan, V. Hautamäki, S. M. Siniscalchi, E. Khoury,
T. Kurki, T. Kinnunen, and C.-H. Lee
"Dialect Levelling in Finnish: A Universal Speech
Attribute Approach"
in Proc. of INTERSPEECH,
pp. 2165–2169, Singapore, 2014.

Dialect Levelling in Finnish: A Universal Speech Attribute Approach

Hamid Behravan[1,4], Ville Hautamäki[1], Sabato Marco Siniscalchi[2,5], Elie Khoury[3], Tommi Kurki[4], Tomi Kinnunen[1] and Chin-Hui Lee[5]

[1]School of Computing, University of Eastern Finland, Finland
[2]Faculty of Architecture and Engineering, University of Enna "Kore", Italy
[3]Idiap Research Institute, Switzerland
[4]School of Languages and Translation Studies, University of Turku, Finland
[5]School of ECE, Georgia Institute of Technology, USA

behravan@cs.uef.fi

Abstract

We adopt automatic language recognition methods to study dialect levelling — a phenomenon that leads to reduced structural differences among dialects in a given spoken language. In terms of dialect characterisation, levelling is a nuisance variable that adversely affects recognition accuracy: the more similar two dialects are, the harder it is to set them apart. We address levelling in Finnish regional dialects using a new SAPU (Satakunta in Speech) corpus containing material from Satakunta (South-Western Finland) between 2007 and 2013. To define a compact and universal set of sound units to characterize dialects, we adopt speech attributes features, namely manner and place of articulation. It will be shown that speech attribute distributions can indeed characterise differences among dialects. Experiments with an i-vector system suggest that (1) the attribute features achieve higher dialect recognition accuracy and (2) they are less sensitive against age-related levelling in comparison to traditional spectral approach.

Index Terms: Finnish language, dialect levelling, social factors, native dialects, dialect detection, attributes detection, i-vector modelling.

1. Introduction

Dialect refers to linguistic variations of a standard spoken language [1]. Over the years, stereotypical differences among dialects of the same spoken language have become smoother and smoother [2] due several co-occurring factors such as language standardisation, industrialisation (increased people mobility) and modernisation (mass media diffusion) [2, 3]. The reduction of peculiar differences among dialects is referred to as *levelling* [4, 5]. Levelling is a common phenomenon in languages. For example, the effect of levelling due to language standardisation can be seen in the phoneme /d/ that is a standard variant in Finnish and also in Pori and Rauma dialects. It has dialectal phonemes in all other dialects of this region, but shows levelling for instance in Honkilahti. That is, Honkilahti has been influenced by the standard Finnish.

In fact, spoken sentences produced by speakers of regional dialects may still be characterised by dialect-specific cues, but levelling weakens such cues, making automatic dialect recognition a hard task. In our task of dialect characterisation, we consider levelling to be a nuisance factor to be compensated for. The problem is analogous to foreign accent recognition [6, 7, 8, 9], where the speakers's second language (L2) masks his mother tongue (L1).

Automatic dialect recognition is traditionally treated as a language recognition problem. State-of-the-art language recognition techniques, either *acoustic* [10, 11] or *phonotactic* [12, 13, 14] ones, can be applied to regional dialect recognition [15, 16]. Although the former techniques have recently proven to attain better language recognition performance [17] by embedding acoustic spectral features within the i-vector framework, there are linguistic and paralinguistic cues (e.g., speaker's age, vocal tract articulators) which can be used for dialect discrimination. We, therefore, propose an articulatory-motivated features with an i-vector method. More specifically, so-called *automatic speech attribute transcription* (ASAT) approach [18, 19, 20, 21, 22] is adopted in order to generate the features of interest for this work, and a bank of detectors is built to detect the presence of speech attributes in a given segment of speech. The speech attributes chosen represent a language-universal set of units, namely manner and place of articulation classes, detected with the help of artificial neural networks.

Indeed, we have already demonstrated that by coupling universal attribute detectors and a state-of-the-art i-vector approach, Finnish foreign accents can be accurately discriminated [7]. Furthermore, ASAT speech attributes have been proven useful in automatic language recognition tasks [23] and cross-language recognition of "unseen" languages using minimal training data from the target languages [24]. The universality of our speech attributes can be better appreciated by thinking of that our detectors were *not* built using ad-hoc Finnish material. In fact, the set of attribute detectors is one used to carry out the independent language recognition experiments reported in [24].

A recently-collected SAPU (Satakunta in Speech) corpus is used to validate our approach. The SAPU corpus includes 8 Finnish sub-dialects or regional dialects and hundreds of speakers. The SAPU Corpus was collected in an interview setting, where subjects interacted with the interviewer in a conversational way. However, interviewer's speech is included in the recording, so needed to be removed by using speaker diarization.

We study three levelling factors: age, gender and place of birth. We first investigate how levelling affects dialect recognition accuracy. Then, the strength of levelling as a function of the speaker's age is investigated. We hypothesize that younger speakers might have lost some of the stereotypical features of their regional dialect, which might still be clear in older speakers of the same region.

2. System description

2.1. Attribute detection

The collection of information embedded into the speech signal, referred to as attributes of speech, also includes the speaker profile encompassing gender, accent, emotional state and other speaker characteristics, which may come useful to automatically uncover the speaker's dialect in a spoken utterance. Indeed, speakers from different regions of a same country may pronounce/produce nasal sounds with diverse acoustic characteristics. Moreover, speakers may also use speech patterns (i.e., conventions of vocabulary, pronunciation, grammar, and usage of certain words) that differ from region to region of the same nation. In this work, the speech attributes of interest are mainly phonetic features, and a bank of speech attribute detectors is built to automatically extract phonetic information from the speech signal. Specifically, five manner of articulation classes (**glide, fricative, nasal, stop, and vowel**), nine place of articulation classes (**coronal, dental, glottal, high, labial, low, mid, retroflex, and velar**), and **voicing** are used. Those attributes could be identified from a particular language and shared across many different languages, so they could also be used to derive a universal set of speech units. Furthermore, data-sharing across languages at the acoustic phonetic attribute level is naturally facilitated by using these attributes, and reliable language-independent acoustic parameter estimation can be anticipated [24].

Each detector is individually designed for modelling a particular speech attribute, and it is built employing three single hidden layer feed-forward multi-layer perceptrons (MLPs) hierarchically organised as described in [25]. These detectors are trained on sub-band energy trajectories that are extracted with a 15 band uniform Mel-frequency filterbank. For each critical band a window of $310ms$ centred around the frame being processed is considered and split in two halves: left-context and right-context [26]. Two independent front-end MLPs ("lower nets") are designed on those two halves and deliver left- and right-context speech attribute posterior probabilities, respectively. Usually, the discrete cosine transform is applied to the input of these lower nets to reduce dimensionality. The outputs of the two lower nets are then sent to the third MLP that acts as a merger and gives the attribute-state posterior probability of the target speech attribute.

Overall, each detector converts an input speech signal into a time series which describes the level of presence (or level of activity) of a particular property of an attribute, or event, in the input speech utterance over time. By using MLPs, the posteriori probability of the particular attribute, given the speech signal, is computed. Articulatory detectors are trained using the same corpus as in [7].

2.2. I-vector modelling

I-vector modelling is rooted on Bayesian *factor analysis* technique which forms a low-dimensional *total variability space* containing both speaker and channel variabilities [27]. In this approach, *universal background model* (UBM), which is a M-component Gaussian mixture model parameterised by $\{w_m, \mathbf{m}_m, \mathbf{\Sigma}_m\}, m = 1, \ldots, M$, where we have mixture weight, mean vector and covariance matrix, respectively. We restrict the covariance matrices to be diagonal. The i-vector model is defined for the UBM component m as [27]:

$$\mathbf{s}_m = \mathbf{m}_m + \mathbf{V}_m \mathbf{y} + \epsilon_m, \tag{1}$$

where $\mathbf{V}_m$ is the sub-matrix of the total variability matrix, $\mathbf{y}$ is the latent vector, called an i-vector, ϵ_m is the residual term and $\mathbf{s}_m$ is the m'th sub-vector of the utterance dependent supervector. The ϵ_m is distributed as $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_m)$, where $\mathbf{\Sigma}_m$ is a diagonal matrix. Given all these definitions, posterior density of the $\mathbf{y}$, given the sequence of observed feature vectors, is Gaussian. Expectation of the posterior is the extracted i-vector. Hyperparameters of the i-vector model, $\mathbf{m}_m$ and $\mathbf{\Sigma}_m$, are copied directly from the UBM and $\mathbf{V}_m$ are estimated by the expectation maximization (EM) algorithm from the same corpus as is used to estimate the UBM.

The *cosine scoring* method is used to compare $\mathbf{w}_{\text{test}}$ and $\mathbf{w}_{\text{target}}$ i-vectors [27]. Cosine score of two i-vectors $\mathbf{w}_{\text{test}}$ and $\mathbf{w}_{\text{target}}$ is computed as their inner product $\langle \mathbf{w}_{\text{test}}, \mathbf{w}_{\text{target}} \rangle$, as

$$\text{s}(\mathbf{w}_{\text{test}}, \mathbf{w}_{\text{target}}) = \frac{\hat{\mathbf{w}}_{\text{test}}^{\text{T}} \ \hat{\mathbf{w}}_{\text{target}}}{\|\hat{\mathbf{w}}_{\text{test}}\| \ \|\hat{\mathbf{w}}_{\text{target}}\|}, \tag{2}$$

where $\hat{\mathbf{w}}_{\text{test}}$ is

$$\hat{\mathbf{w}}_{\text{test}} = \mathbf{A}^{\text{T}} \mathbf{w}_{\text{test}}, \tag{3}$$

and $\mathbf{A}$ is the *heteroscedastic linear discriminant analysis* (HLDA) projection matrix [28] estimated from all training utterances. Further, $\hat{\mathbf{w}}_{\text{target}}$ is defined for a given dialect as,

$$\hat{\mathbf{w}}_{\text{target}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \hat{\mathbf{w}}_{id}, \tag{4}$$

where N_d is the number of training utterances in dialect d and $\hat{\mathbf{w}}_{id}$ is the projected i-vector of training utterance i from dialect d, computed the same way as (3). Obtaining $\{s_d, d = 1, \ldots, N\}$ scores for test utterances of dialect d, and total number of targeted models, N, scores are post-processed as [29]:

$$s'(d) = \log \frac{\exp(s_d)}{\frac{1}{N-1} \sum_{k \neq d} \exp(s_k)} \tag{5}$$

$s'(d)$ is the detection log-likelihood ratio and is used in the detection task.

3. Evaluation setup

3.1. Corpora

SAPU (Satakunta in Speech) corpus have been used to perform a series of experiments in this study. The data recorded in Satakunta, in southwestern Finland 2007-2013, in an interview setting. The topics were related to informants life and home region. Currently, the corpus consists of 282 recordings (231 hours 31 minutes)[1].

Satakunta region is divided into two distinctive dialectal regions, Southwestern dialects and the dialects of Häme. For our purposes, we selected 8 dialects — Luvia, Kokemäki, Honkilahti, Pori, Eurajoki, Rauma, Harjavalta, and Ulvila — with enough available data. All the audio files were partitioned into wave files of 30 seconds in duration, and downsampled to 8 kHz sampling rate. Table 1 shows the train and test files distributions within each dialects. There is no speaker overlap between training and test files.

[1]Corpus is located at the University of Turku the Syntax Archives server and is available by request.

Table 1: Training and test files distribution in the SAPU corpus.

Dialect	#Train files	#Test files	#Speakers
Luvia	386	315	31
Kokemäki	689	438	27
Honkilahti	845	413	24
Pori	341	289	15
Eurajoki	256	237	13
Rauma	237	64	9
Harjavalta	66	65	4
Ulvila	113	36	4

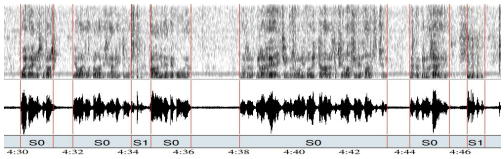


Figure 1: Speaker diarization scheme. S0 is the majority class (interviewee) and S1 is the minority class (interviewer).

3.2. Diarization

Two speakers are generally involved in the interviews. Speaker diarization [30] aims at 1) segmenting the audio stream into speech utterances, and 2) grouping all utterances belonging to the same speaker. In this study, diarization is mainly inspired by [31]. After noise reduction[2], a bidirectional audio source segmentation is applied using both *generalized likelihood ratio* (GLR) [32] and *Bayesian information criterion* (BIC). The resulting segments serve as an initial set of clusters that feed the clustering process.

This clustering is a variant of the Gaussian Mixture Model (GMM) system widely used for speaker recognition. The universal background model (UBM) is trained using all speech utterances from all recordings. To cope with short duration clusters, only 32 Gaussian components are used. Finally, the major cluster is selected and used for dialect recognition.

Fig. 1 shows the diarization scheme for a sample audio file. For this example, S0 is the majority class (interviewee) and S1 is the minority class (interviewer).

3.3. Measurement metrics

System performance is reported in terms of *equal error rate* (EER) and average detection cost (C_{avg}). EER corresponds to the operating point where false alarm and miss probabilities are equal. We report averaged EER across dialect-specific EERs. C_{avg} is defined as,

$$C_{\text{avg}} = \frac{1}{J}\sum_{j=1}^{M} C_{\text{DET}}(L_j), \tag{6}$$

where $C_{\text{DET}}(L_j)$ is the detection cost for subset of test segments trials for which the target dialect is L_j and J is the number of target languages. The per target dialect cost is computed as,

$$\begin{aligned} C_{\text{DET}}(L_j) &= C_{\text{miss}} P_{\text{tar}} P_{\text{miss}}(L_j) \\ &+ C_{\text{fa}}(1-P_{\text{tar}})\frac{1}{J-1}\sum_{k\neq j} P_{\text{fa}}(L_j, L_k) \end{aligned} \tag{7}$$

The miss probability (or false rejection rate) is denoted by P_{miss}, i.e., a test segment of dialect L_i is rejected as being in that dialect. On the other hand $P_{\text{fa}}(L_i, L_k)$ denotes the probability when a test segment of dialect L_k is accepted as being in dialect L_i. It is computed for each target/non-target dialect pairs. C_{miss} and C_{fa} are costs of making errors and both were set to 1. P_{tar} is the prior probability of a target dialect and was set to 0.5.

4. Results

4.1. Finnish dialect detection

We introduce speech attribute based systems in dialect recognition task and contrast it with baseline shifted delta cepstra and Mel frequency cepstral coefficients (SDC+MFCC), and single attribute (manner or place) system in Table 2. The parameters and combination (SDC and MFCC) were optimised in [6]. We also present results for attributes stacked across multiple frames. That is, we stack the estimated attribute feature vectors (either place or manner) across K neighboring frames to create a high-dimensional context feature vector. As discussed in detail in [7], the dimensionality of the context vector is reduced with principal component analysis (PCA). The PCA bases are trained from the same utterances as the universal background model (UBM), with 99% variance retained by the leading eigenvectors. In this work, we found that the PCA of context size $C = 10$ gives the best result on attributes. The PCA manner outperforms the baseline SDC+MFCC by 25% relative improvement considering C_{avg}. It also outperforms single manner and place attributes by 15% and 23% relative improvements, respectively. The place PCA is found not to be effective. This seems to contradict our earlier finding on another corpus [7]. While the exact reason is presently unknown, we note that the automatically determined PCA dimensionality for place attributes is smaller than in [7].

Literature of regional automatic dialect recognition is limited. In a study by DeMarco and Cox [15], SDC based i-vector system was used to classify fourteen British accents resulting 32% Id_{err}, which is comparable to 36% Id_{err} in Table 2. Later they improved the error rate to 19% by a very large scale fusion [16].

Table 2: Summary of results and compared against baseline spectral system, results are shown in average EER (Avg EER), C_{avg} and identification error rate (Id_{err}). C and d are context size and feature dimensionality, respectively.

Features (dimensionality)	Avg EER (%)	$C_{\text{avg}} \times 100$	Id_{err} (%)
SDC+MFCC (56)	14.20	5.31	36.08
Manner (18)	13.47	4.76	29.88
Place (27)	16.12	5.18	34.16
Manner+Place (45)	13.67	4.58	29.16
PCA Manner (C=10,d=30)	**12.52**	**4.00**	**29.01**
PCA Place (C=10,d=13)	17.60	5.64	37.65

[2]http://www1.icsi.berkeley.edu/Speech/papers/qio/

4.2. Levelling analysis

Here, we will further analyze the averaged detection results in terms of age groups. Fig. 2 presents the results per age group; that is, we choose a subset of original trials constrained to a given age group. We notice that the dialect in the younger age groups is considerably more difficult recognize than in the older age groups. The result indicates that the dialect of younger speakers has levelled. On the other hand, PCA manner considerably outperforms baseline SDC+MFCC for the youngest age group. It implies that attribute system is robust against the age related levelling for younger speakers.

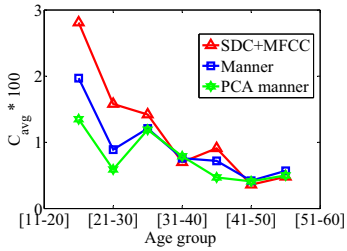


Figure 2: C_{avg} at different age groups.

We investigated more closely the cuts in the age group 11-20 that are correctly recognized by PCA Manner but incorrectly recognized by the spectral system, totalling 83 cuts from 19 different individuals. We show the example in Fig. 3, of one 30 seconds cut from a female speaker who is from Honkilahti municipality, however, in this cut she is recognized as being from Rauma by the spectral system. In this example, she says "*mum mielest se*" (in my opinion), where we notice word-final /n/ assimilated to bilabial nasal /m/. This would not happen in the Pori region dialects. Such an assimilation is typical for all the Southwestern Dialects (including Luvia and when preceded by bilabial phoneme /m/ Honkilanti). Of three detector scores per attribute we show here only the target score for clarity. We notice the nasal component is strong in the middle /m/, where dialectal difference shows.

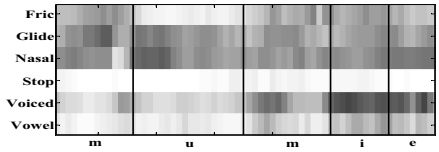


Figure 3: Target detection scores for the Manner of articulation detectors shown for the portion of "mum mielest se" (in my opinion).

It is interesting to see how much the attribute detection errors affect the dialect recognition performance for age group between 11 to 20 years old. Table 3 shows the confusion matrix of PCA manner system for this age group. Honkilahti is often misclassified as being Kokemäki, Pori, Eura; and Kokemäki being often misclassified as Ulvila. For Honkilahti dialects, the misclassification comes from the common prosodic features. on the other hand, Ulvila and Kokemäki are both Häme dialects.

Fig.4 shows how C_{avg} is affected by gender and region of birth for different systems. The dialectal differences of females is easier to recognize than for males. Similar to age analysis, PCA manner outperforms baseline SDC+MFCC and manner system. According to [33], various phonological and lexical forms and the syntactic-pragmatic features identified occur more often in women's than men's speech. Taking region of birth, results disagree with the common notion that those living in their home region have stronger dialects than those who have migrated from their home region. According to [34], language use of some migrated speakers show great situational variation. While there are always significant differences between the speakers of the same community, sometimes migrated speakers may speak even more dialectically. This kind of dialectal boosting appears specially in emphatic and affective occasions, when speaker talks with another person from the same region about the home region and people living there. The recordings of this corpus were recorded by the assistants born and raised in the same region.

Table 3: Confusion matrix of PCA manner system for age group between 11 and 20 years old. (There are no Eue, Rau and Har test utterances available for this age group.)

		Predicted label							
		Luv	Kok	Hon	Por	Eur	Rau	Har	Ulv
True label	Luv	35	3	7	4	6	10	1	1
	Kok	21	30	18	23	16	2	23	27
	Hon	23	41	168	48	57	23	24	20
	Por	7	0	8	23	8	0	1	6
	Ulv	2	7	7	3	0	1	1	9

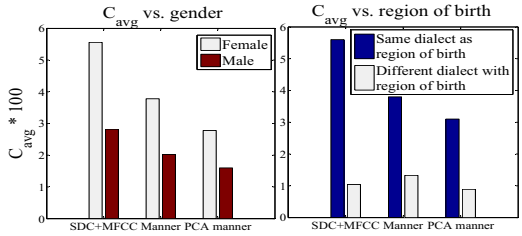


Figure 4: C_{avg} per gender and region of birth.

5. Conclusion

In this paper, we experimented with regional dialect recognition task. In terms of absolute error rates, it was shown to be a difficult task. There are two major sources of difficulty, differences between regional dialects are very small and the dialects are affected by the levelling phenomenon. Three levelling effects, age, gender and region of birth were studied in this paper. We showed that manner of articulation based recognition system can efficiently compensate the age levelling effect in Finnish dialect recognition. Furthermore, adding context information to manner attributes considerably improved the results.

6. Acknowledgements

This work was partly supported by Academy of Finland (projects 253000, 253120 and 283256) and Kone foundation.

7. References

[1] D. Britain, *Geolinguistics and linguistic diffusion*. Sociolinguistics: International Handbook of the Science of Language and Society, 2005.

[2] P. Kerswill and A. Williams, "Mobility and social class in dialect levelling: evidence from new and old towns in England," in *Dialect and migration in a changing Europe.*, Peter Lang, Frankfurt, 2000, pp. 1–13.

[3] E. Torgersen and P. Kerswill, "Internal and external motivation in phonetic change: Dialect levelling outcomes for an English vowel shift," *Journal of Sociolinguistics*, vol. 8, pp. 23–53, 2004.

[4] P. Kerswill, "Dialect levelling and geographical diffusion in British English," in *Social dialectology: in honour of Peter Trudgill*. Amsterdam: Benjamins, 2003, pp. 223–243.

[5] P. Kerswill and A. Williams, "Dialect levelling: change and continuity in Milton Keynes, Reading and Hull." in *Urban voices. Accent studies in the British Isles*, Arnold, London, 1999, pp. 141–162.

[6] H. Behravan, V. Hautamäki, and T. Kinnunen, "Foreign accent detection from spoken Finnish using i-vectors," in *INTERSPEECH*, Lyon, France, August 2013.

[7] H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C.-H. Lee, "Introducing attribute features to foreign accent recognition," in *Acoustics, Speech and Signal Processing, 2014. ICASSP 2014. IEEE International Conference on*. IEEE, 2014.

[8] O. Scharenborg, M. J. Witteman, and A. Weber, "Computational modelling of the recognition of foreign-accented speech," in *INTERSPEECH*. ISCA, 2012.

[9] M. Bahari, R. Saeidi, H. Van hamme, and D. van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector, Guassian posterior probability for spontaneous telephone speech," in *ICASSP*, Vancouver, Canada, 2013.

[10] N. F. Chen, W. Shen, and J. P. Campbell, "A linguistically-informative approach to dialect recognition using dialect-discriminating context-dependent phonetic models," in *Acoustics, Speech and Signal Processing, 2014. ICASSP 2014. IEEE International Conference on*, 2010, pp. 5014–5017.

[11] F. Biadsy, J. Hirschberg, and M. Collins, "Dialect recognition using a phone-GMM-supervector-based SVM kernel." in *INTERSPEECH*. ISCA, 2010, pp. 753–756.

[12] N. F. Chen, W. Shen, J. P. Campbell, and P. A. Torres-Carrasquillo, "Informative dialect recognition using context-dependent pronunciation modelling." in *Acoustics, Speech and Signal Processing, 2011. ICASSP 2011. IEEE International Conference on*. IEEE, 2011, pp. 4396–4399.

[13] F. Biadsy, H. Soltau, L. Mangu, J. Navratil, and J. Hirschberg, "Discriminative phonotactics for dialect recognition using context-dependent phone classifiers," in *Odyssey*, 2010, p. 44.

[14] S. Sinha, A. Jain, and S. S. Agrawal, "Speech processing for Hindi dialect recognition," *Advances in Intelligent Systems and Computing*, vol. 264, pp. 161–169, 2014.

[15] A. DeMarco and S. J. Cox, "Iterative classification of regional British accents in i-vector space," in *MLSLP*, 2012, pp. 1–4.

[16] A. DeMarco and S. J. Cox., "Native accent classification via i-vectors and speaker compensation fusion," in *INTERSPEECH*, 2013, pp. 1472–1476.

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

[18] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *INTERSPEECH*, Jeju Island, Korea, 2004, pp. 109–112.

[19] I. Bromberg, Q. Fu, J. Hou, J. Li, C. Ma, B. Matthews, A. Moreno-Daniel, J. Morris, S. M. Siniscalchi, Y. Tsao, and Y. Wang, "Detection-based ASR in the automatic speech attribute transcription project," in *INTERSPEECH*, Antwerp, Belgium, 2007, pp. 1829–1832.

[20] C.-Y. Chiang, S. M. Siniscalchi, Y.-R. Wang, S.-H. Chen, and C.-H. Lee, "A study on cross-language knowledge integration in Mandarin LVCSR," in *Proc. ISCSLP*, HONG KONG, Dec. 2012, pp. 315–319.

[21] C.-Y. Chiang, S. M. Siniscalchi, , S.-H. Chen, and C.-H. Lee, "Knowledge integration for improving performance in LVCSR," in *Proc. INTERSPEECH*, Lyon, France, Aug. 2013, pp. 1786–1790.

[22] C.-H. Lee and S. M. Siniscalchi, "An information-extraction approach to speech processing: Analysis, detection, verification, and recognition," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1089–1115, 2013.

[23] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Universal attribute characterization of spoken languages for automatic spoken language recognition," *Computer Speech & Language*, vol. 27, no. 1, pp. 209–227, 2013.

[24] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target specific training data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 875–887, 2012.

[25] S. M. Siniscalchi, T. Svendsen, and C.-H. Lee, "Toward a detector-based universal phone recognizer," in *Proc. ICASSP*, Las Vegas, NV, USA, Mar./Apr. 2008, pp. 4261–4264.

[26] P. Schwarz, P. Matějaka, and J. Cernock, "Hierarchical structures of neural networks for phoneme recognition," in *ICASSP*, Toulouse, France, 2006, pp. 325–328.

[27] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 788–798, 2011.

[28] M. Loog and R. P. W. Duin, "Linear dimensionality reduction via a heteroscedastic extension of LDA: The chernoff criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 732–739, 2004.

[29] N. Brummer and D. Van Leeuwen, "On calibration of language recognition scores," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, June 2006, pp. 1–8.

[30] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 356–370, Feb 2012.

[31] E. Khoury, C. Senac, and J. Pinquier, "Improved speaker diarization system for meetings," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4097–4100.

[32] C. Barras, X. Zhu, S. Meignier, and J. Gauvain, "Multistage speaker diarization of broadcast news," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1505–1512, Sept 2006.

[33] R. Lakoff, *Language and woman's place*, ser. Harper colophon books. Harper & Row, 1975.

[34] P. Nuolijrvi, *Kieliyhteisn vaihto ja muuttajan identiteetti.*, ser. Tietolipas. Helsinki: SKS., 1986.

Paper **IV**

H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C.-H. Lee
"i-Vector Modeling of Speech Attributes for Automatic Foreign Accent Recognition"
IEEE/ACM Transactions on Audio, Speech, and Language Processing,
24, 29–41 (2016).

i-Vector Modeling of Speech Attributes for Automatic Foreign Accent Recognition

Hamid Behravan, *Member IEEE*, Ville Hautamäki, *Member IEEE*, Sabato Marco Siniscalchi, *Member IEEE*, Tomi Kinnunen, *Member IEEE*, and Chin-Hui Lee, *Fellow IEEE*

***Abstract*—We propose a unified approach to automatic foreign accent recognition. It takes advantage of recent technology advances in both linguistics and acoustics based modeling techniques in automatic speech recognition (ASR) while overcoming the issue of a lack of a large set of transcribed data often required in designing state-of-the-art ASR systems. The key idea lies in defining a common set of fundamental units "universally" across all spoken accents such that any given spoken utterance can be transcribed with this set of "accent-universal" units. In this study, we adopt a set of units describing manner and place of articulation as speech attributes. These units exist in most spoken languages and they can be reliably modeled and extracted to represent foreign accent cues. We also propose an i-vector representation strategy to model the feature streams formed by concatenating these units. Testing on both the Finnish national foreign language certificate (FSD) corpus and the English NIST 2008 SRE corpus, the experimental results with the proposed approach demonstrate a significant system performance improvement with p-value < 0.05 over those with the conventional spectrum-based techniques. We observed up to a 15% relative error reduction over the already very strong i-vector accented recognition system when only manner information is used. Additional improvement is obtained by adding place of articulation clues along with context information. Furthermore, diagnostic information provided by the proposed approach can be useful to the designers to further enhance the system performance.**

***Index Terms*—Attribute detectors, i-vector system, Finnish corpus, English corpus.**

I. Introduction

AUTOMATIC foreign accent recognition is the task of identifying the mother tongue (L1) of non-native speakers given an utterance spoken in a second language (L2) [1]. The task attracts increasing attention in the speech community because accent adversely affects the accuracy of conventional *automatic speech recognition* (ASR) systems (e.g., [2]). In fact, most existing ASR systems are tailored to native speech only, and recognition rates decrease drastically when words or sentences are uttered with an altered pronunciation (e.g., foreign accent) [3]. Foreign accent variation is a nuisance factor that adversely affects automatic speaker and language recognition systems as well [4], [5]. Furthermore, foreign accent recognition is a topic of great interest in the areas of intelligence and security, including immigration screening and border control sites [6]. It may help officials detect a fake passport by verifying whether a traveler's spoken foreign accent corresponds to accents spoken in the country he claims he is from [6]. Finally, connecting customers to agents with similar foreign accent in targeted advertisement applications may help create a more user-friendly environment [7].

It is worth noting that *foreign accents* differ from *regional accents* (dialects), since the deviation from the standard pronunciation depends upon the influence that L1 has on L2 [8]. Firstly, non-native speakers tend to alter some phone features when producing a word in L2 because they only partially master its pronunciation. To exemplify, Italians often do not aspirate the /h/ sound in words such as *house*, *hill*, and *hotel*. Moreover, non-native speakers can also replace an unfamiliar phoneme in L2 with the one considered as the closest in their L1 phoneme inventory. Secondly, there are several degrees of foreign accent for the same native language influence according to L1 language proficiency of the non-native speaker [9], [10]: non-native speaker learning L2 at an earlier age can better compensate for their foreign accent factors when speaking in L2 [11].

In this study, we focus on automatic L1 detection from spoken utterances with the help of statistical pattern recognition techniques. In the following, we give a brief overview and current state-of-the-art methods before outlining our contributions. It is common practice to adopt automatic *language recognition* (LRE) techniques to the foreign accent recognition task. Indeed, the goal of an LRE system is to automatically detect the spoken language in an utterance, which we can parallel with that of detecting L1 in an L2 utterance. Automatic LRE techniques can be grouped into to main categories: *token-based* (a.k.a, *phonotactic*) and *spectral-based* ones. In the token-based approach, discrete units/tokens, such as phones, are used to describe any spoken language. For example, parallel phone recognition followed by language modeling (PPRLM) [12] approach employs a bank of phone recognizers

Manuscript received December 27, 2014; revised June 15, 2015; accepted September 28, 2015. This project was partially supported by the Academy of Finland projects 253120, 253000 and 283256, Finnish Scientific Advisory Board for Defence (MATINE) project nr. 2500M-0036 and Kone Foundation - Finland. Dr. Hautamäki and Dr. Siniscalchi were supported by the Nokia Visiting Professor Grants 201500062 and 201600008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mohamed Afify.

H. Behravan is with the School of Computing, University of Eastern Finland, Joensuu, Finland. E-mail: behravan@cs.uef.fi

V. Hautamäki is with the School of Computing, University of Eastern Finland, Joensuu, Finland. E-mail: villeh@cs.uef.fi

S. M. Siniscalchi is with the Department of Computer Engineering, Kore University of Enna, Enna, Italy, and with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: marco.siniscalchi@unikore.it

T. Kinnunen is with the School of Computing, University of Eastern Finland, Joensuu, Finland. E-mail: tkinnu@cs.uef.fi

C.-H. Lee is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: chl@ece.gatech.edu

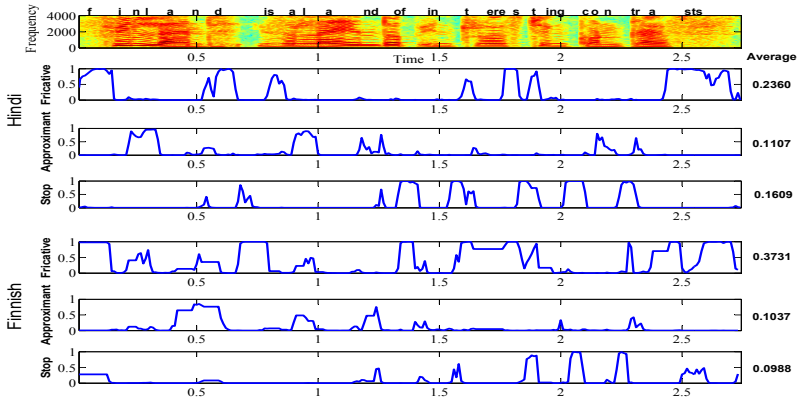


Fig. 1: An example showing the detection score differences in the three selected attributes from a Hindi and a Finnish speaker. Both speakers utter the same sentence 'Finland is a land of interesting contrasts'. Speech segments are time-aligned with dynamic time warping (DTW). The Finnish speaker shows higher level of activity in fricative in comparison to the Hindi speaker. However, in the Hindi speech utterance, the level of activity in stop is higher than in the Finnish utterance.

to convert each speech utterance into a string of tokens. In the spectral-based approach a spoken utterance is represented as a sequence of short-time spectral feature vectors. These spectral vectors are assumed to have statistical characteristics that differ from one language to another [13], [14]. Incorporating temporal contextual information to the spectral feature stream has been found useful in the language recognition task via the so-called *shifted-delta-cepstral* (SDC) features [15]. The long-term distribution of language-specific spectral vectors is modeled, in one form or another, via a language- and speaker-independent universal background model (UBM) [16]. In the traditional approaches [16], [17], language-specific models are obtained via UBM adaptation while the modern approach utilizes UBMs to extract low-dimensional *i-vectors* [18]. I-vectors are convenient for expressing utterances with varying numbers of observations as a single vector that preserves most utterance variations. Hence, issues such as session normalization are postponed to back-end modeling of i-vector distributions.

Table I shows a summary of several studies on foreign accent recognition. In [1], the accented speech is characterized using acoustic features such as frame power, zero-crossing rate, LP reflection coefficients, autocorrelation lags, log-area-ratios, line-spectral pair frequencies and LP cepstrum coefficients. 3-state hidden Markov models (HMMs) with a single Gaussian density were trained from these features and evaluated on spoken American English with 5 foreign accents reporting 81.5% identification accuracy. The negative effects of non-native accent in ASR task were studied in [19]. Whole-word and sub-word HMMs were trained on either native accent utterances or a pool of native and non-native accent sentences. The use of phonetic transcriptions for each specific accent improved speech recognition accuracy. An accent dependent parallel phoneme recognizer was developed in [20] to discriminate native Australian English speakers and two migrant speaker groups with foreign accents, whose L1's were either Levantine Arabic or South Vietnamese. The best average accent identification accuracies of 85.3% and 76.6% for accent pair and three accent class discrimination tasks were reported, respectively. A text-independent automatic accent classification system was deployed in [5] using a corpus representing five English speaker groups with native American English, and English spoken with Mandarin Chinese, French, Thai and Turkish accents. The proposed system was based on stochastic and parametric trajectory models corresponding to the sequence of points reflecting movements in the speech production caused by coarticulation. This system achieved an accent classification accuracy of 90%.

All the previous studies used either suprasegmental modeling, in terms of trajectory model or prosody, or phonotactic modeling to recognize non-native accents. Recently, spectral features with i-vector back-end were found to outperform phonotactic systems in language recognition [18]. Spectral features were first used by [21] in a L1 recognition task. The non-native English speakers were recognized using multiple spectral systems, including i-vectors with different back-ends [21], [23]. The i-vector system outperformed other methods most of the time, and spectral techniques based on i-vector model are thus usually adopted for accent recognition. The lack of large amount of transcribed accent-specific speech data to train high-performance acoustic phone models hinders the deployment of competitive phonotactic foreign accent recognizers. Nonetheless, it could be argued that phonotactic methods would provide valuable results that are informative to humans [24]. Thus, a unified foreign accent recognition framework that gives the advantages of the subspace modeling techniques without discharging the valuable information provided by the phonotactic-based methods is highly desirable.

The *automatic speech attribute transcription* (ASAT) framework [25], [26], [27] represents a natural environment to make these two above contrasting goals compatible, and is adopted here as the reference paradigm. The key idea of ASAT is

TABLE I: Summary of the previous studies on foreign accent recognition and the present study.

Study	Spoken language	#accents	#speakers	#utterances	Features	Model
Hansen and Arslan [1]	American English	4	27	N/A	Prosodic	HMM
Teixeira et al. [19]	British English	5	20	20	Phonotactic	HMM
Kumpf and King [20]	Australian English	3	67	3950	Phonotactic	HMM
Angkititraku and Hansen [5]	English	5	179	N/A	Phoneme sequence	Trajectory-model
Bahari et al. [21]	English	5	265	359	Spectral	GMM supervector
Behravan et al. [22], [10]	Finnish	9	450	1973	Spectral	i-vector modeling
Present study	Finnish (FSD)	8	415	1644	Attributes	i-vector modeling
Present study	English (NIST)	7	348	1262	Attributes	i-vector modeling

to use a compact set of speech attributes, such as *fricative*, *nasal* and *voicing* to compactly characterize any L2 spoken sentence independently of the underlying L1 native language. A bank of data-driven detectors generates attribute posterior probabilities, which are in turn modeled using an i-vector back-end, treating the attribute posteriors as acoustic features. A small set of speech attributes suffices for a complete characterization of spoken languages, and it can therefore be useful to discriminate accents [28]. For example, some sister languages, e.g., Arabic spoken in Syria and Iraq, only have subtle differences that word-based discrimination usually does not deliver good results. In contrast, these differences naturally arise at an attribute level and can help foreign accent recognition. Robust universal speech attribute detectors can be designed by sharing data among different languages, as shown in [29], and that bypasses the lack of sufficient labeled data for designing ad-hoc tokenizers for a specific L1/L2 pair. Indeed, the experiments reported in this work concern detecting Finnish and English foreign accented speech, even though the set of attribute detectors was originally designed to address phone recognition with minimal target-specific training data [29]. Although speech attributes are shared across spoken languages, the statistics of the attributes can differ considerably from one foreign accent to another, and these statistics improve discrimination [30]. This can be appreciated by visually inspecting Figure 1, which shows attribute detection curves from Finnish and Hindi speakers. Although both speakers uttered the same sentence, namely "Finnish is a land of interesting contrasts," differences between corresponding attribute detection curves can be observed: (i) the fricative detection curve tends to be more active (i.e. stays close to 1) in Finnish speaker than in Hindi, (ii) the stop detection curve for the Hindi speaker more often remains higher (1 or close to 1) than that for the Finnish speaker, (iii) approximant detection curve seem instead to show similar level of activity for both speakers.

In this work, we significantly expand our preliminary findings on automatic accent recognition [31] and re-organize our work in a systematic and, self-contained form that provides a convincing case why universal speech attributes are worthwhile of further studies in accent characterization. The key experiments, not available in [31], can be summarized as follows: (i) we have investigated the effect of *heteroscedastic* linear discriminant analysis (HLDA) [32] dimensionality reduction on the accent recognition performance and compared and contrasted it with linear discriminant analysis (LDA), (ii) we have studied training and test duration effects on the overall system performance, and (iii) we have expanded our initial investigation on Finnish data by including new experiments on English foreign accent. Even if the single components have been individually investigated in previous studies, e.g., [30], [33], [18], the overall architecture (combining the components) presented in this paper, as well as its application to foreign accent recognition, are novel. The key novelty of our framework can be summarized as follows: (i) speech attributes extracted using machine learning techniques are adopted to the foreign accent recognition task for the first time, (ii) a dimensionality reduction approach is used for capturing temporal context and exploring the effect of languages, (iii) the i-vector approach is successfully used to model speech attributes. With respect to point (iii), Diez et al. [34], [35] proposed a similar solution but to address a spoken language recognition task, namely they used log-likelihood ratios of phone posterior probabilities within the i-vector framework. Although Diez et al.'s work has some similarities with ours, there are several implementation differences in addition to the different addressed task: (i) we describe different accents using a compact set of language independent attributes, which overcomes high computational issues caused by high-dimension posterior scores, as mentioned in [34], (ii), we introduce context information by stacking attribute probability vectors together, and we then capture context variability directly in the attribute space, and (iii) we carry out i-vector post-processing to further improve accents discriminability. Moreover, useful diagnostic information can be gathered with our approach, as demonstrated in Section IV-D.

Finally in [22], [10], the authors demonstrated that i-vector modeling using SDCs outperforms conventional Gaussian mixture model - universal background model (GMM-UBM) system in recognizing Finnish non-native accents. The method proposed in [10] is here taken to build a reference baseline system to compare with. We evaluate effectiveness of the proposed attribute-based foreign accent recognition system with a series of experiments on Finnish and English foreign accented speech corpora. The experimental evidence demonstrates that the proposed technique compares favorably with conventional SDC-MFCC with i-vector and GMM-UBM approaches. In order to enhance accent recognition performance of the proposed technique, several configurations have been proposed and evaluated. In particular, it was observed that contextual information helps to decrease recognition error rates.

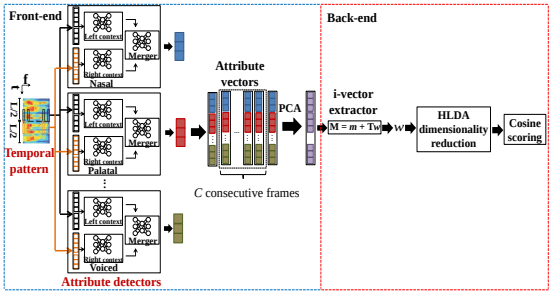


Fig. 2: Block diagram of the proposed system. In the attribute detectors [29], [30], [27], spectral features are fed into left-context and right-context artificial neural networks. A merger then combines the outputs generated by those two neural networks and produce the final attribute posterior probabilities. Principal component analysis (PCA) is then applied on C consecutive frames of these posterior probabilities to create long-term contextual features. We use i-vector approach [33] with cosine scoring [33] to classify target accents.

II. Foreign Accent Recognition

Figure 2 shows the block diagram of the proposed system. The front-end consists of attribute detectors and building long-term contextual features via principal component analysis (PCA). The features created in the front-end are then used to model target foreign accents using a i-vector back-end. In the following, we describe the individual components in detail.

A. Speech attribute extraction

The set of speech attributes used in this work are acoustic phonetic features, namely, five *manner of articulation* classes (**glide, fricative, nasal, stop, and vowel**), and **voicing** together with nine *place of articulation* (**coronal, dental, glottal, high, labial, low, mid, retroflex, velar**). Attributes could be extracted from a particular language and shared across many different languages, so they could also be used to derive a universal set of speech units. Furthermore, data-sharing across languages at the acoustic phonetic attribute level is naturally facilitated by using these attributes, so more reliable language-independent acoustic parameter estimation can be anticipated [29]. In [30], it was also shown that these attributes can be used to compactly characterize any spoken language along the same lines as in the ASAT paradigm for ASR [27]. Therefore, we expect that it can also be useful for characterizing speaker accents.

B. Long-term Attribute Extraction

Each attribute detector outputs the posterior probability for the target class i, $p(H^{(i)}_{\text{target}}|\boldsymbol{f})$, non-target, $p(H^{(i)}_{\text{anti}}|\boldsymbol{f})$, and noise, $p(H^{(i)}_{\text{noise}}|\boldsymbol{f})$, class given a speech frame $\boldsymbol{f}$. As probabilities, they sum up to one for each frame. A feature vector $\boldsymbol{x}$ is obtained by concatenating those posterior probabilities generated by the set of manner/place detectors into a single vector. The final dimension of the feature vector, $\boldsymbol{x}$, is 18 in the manner of articulation case, for example.

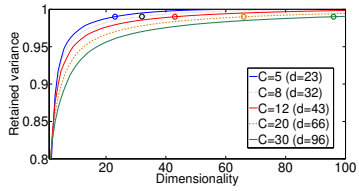


Fig. 3: Remaining variance after PCA. Comparing stacked context sizes (C) 5, 8, 12, 20 and 30 frames for manner attributes. d varies from ∼20 to ∼100, with larger dimensionality assigned to longer context sizes.

Since language and dialect recognizers benefit from the inclusion of long temporal context [36], [16], it is natural to study similar ideas for attribute modeling as well. A simple feature stacking approach is adopted in this paper. To this end, let $\boldsymbol{x}(t) \in \mathbb{R}^n$ denote the 18-dimensional (6 manner attributes $\times$ 3) or 27-dimensional (9 place attributes $\times$ 3) feature attribute vector at frame t. A sequence of $q = 18\,C$ (or $q = 27\,C$, for place) dimensional stacked vectors $\tilde{\boldsymbol{x}}_C(t) = (\boldsymbol{x}(t)^\top, \boldsymbol{x}(t+1)^\top, \ldots, \boldsymbol{x}(t+C-1)^\top)^\top, t = 1, 2, \ldots,$ is formed, where C is the context size, and $\top$ stands for transpose. PCA is used to project each $\tilde{\boldsymbol{x}}_C(t)$ onto the first $d \ll q$ eigenvectors corresponding to the largest eigenvalues of the sample covariance matrix. We estimate the PCA basis from the same data as the UBM and the T-matrix, after VAD, with 50 % overlap across consecutive $\tilde{\boldsymbol{x}}_C(t)$'s. We retain 99 % of the cumulative variance. As Figure 3 indicates, d varies from ∼20 to ∼100, with larger dimensionality assigned to longer context as one expects.

C. I-vector Modeling

I-vector modeling or total variability modeling, forms a low-dimensional *total variability space* that contains spoken content, speaker and channel variability [33]. Given an utterance, a GMM supervector, $\boldsymbol{s}$, is represented as [33],

$$\boldsymbol{s} = \boldsymbol{m} + \mathbf{T}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{m}$ is the utterance- and channel-independent component (the universal background model or UBM supervector), $\mathbf{T}$ is a rectangular low rank matrix and $\boldsymbol{w}$ is an independent random vector of distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. $\mathbf{T}$ represents the captured variabilities in the supervector space. It is estimated by the expectation maximization (EM) algorithm similar to estimating the speaker space in joint factor analysis (JFA) [37], with the exception that every training utterances of a given model is treated as belonging to different class. The extracted i-vector is then the mean of the posterior distribution of $\boldsymbol{w}$.

D. Inter-session Variability Compensation

As the extracted i-vectors contain both within- and between accents variation, we used dimensionality reduction technique

to project the i-vectors onto a space to minimize the within-accent and maximize the between-accent variation. To perform dimensionality reduction, we used *heteroscedastic* linear discriminant analysis (HLDA) [32], which is considered as an extension of linear discriminant analysis (LDA). In this technique, i-vector of dimension n is projected into a p-dimensional feature space with $p < n$, using HLDA transformation matrix denoted by $\mathbf{A}$. The matrix $\mathbf{A}$ is estimated by an efficient row-by-row iteration with EM algorithm as presented in [38].

Followed by HLDA, within-class covariance normalization (WCCN) is then used to further compensate for unwanted intra-class variations in the total variability space [39]. The WCCN transformation matrix, $\mathbf{B}$, is trained using the HLDA-projected i-vectors obtained by Cholesky decomposition of $\mathbf{B}\mathbf{B}^\top = \mathbf{\Lambda}^{-1}$, where a within-class covariance matrix, $\mathbf{\Lambda}$, is computed using,

$$\mathbf{\Lambda} = \frac{1}{L}\sum_{a=1}^{L}\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}_i^a - \overline{\boldsymbol{w}}_a)(\boldsymbol{w}_i^a - \overline{\boldsymbol{w}}_a)^\top, \tag{2}$$

where $\overline{\boldsymbol{w}}_a$ is the mean i-vector for each target accent a, L is the number of target accents and N is the number of training utterances in target accent a. The HLDA-WCCN inter-session variability compensated i-vector, $\hat{\boldsymbol{w}}$, is calculated as,

$$\hat{\boldsymbol{w}} = \mathbf{B}^\top\mathbf{A}^\top\boldsymbol{w}. \tag{3}$$

E. Scoring Against Accent Models

We used *cosine scoring* to measure similarity of two i-vectors [33]. The cosine score, t, between the inter-session variability compensated test i-vector, $\hat{\boldsymbol{w}}_{\text{test}}$, and target i-vector, $\hat{\boldsymbol{w}}_{\text{target}}$, is computed as the dot product between them,

$$t = \frac{\hat{\boldsymbol{w}}_{\text{test}}^\top}{\|\hat{\boldsymbol{w}}_{\text{test}}\|}\frac{\hat{\boldsymbol{w}}_{\text{target}}}{\|\hat{\boldsymbol{w}}_{\text{target}}\|}, \tag{4}$$

where $\hat{\boldsymbol{w}}_{\text{target}}$ is the average i-vector over all the training utterances of the target accent, i.e.

$$\hat{\boldsymbol{w}}_{\text{target}} = \frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{w}}_i, \tag{5}$$

where $\hat{\boldsymbol{w}}_i$ is the inter-session variability compensated i-vector of training utterance i in the target accent.

Obtaining scores $\{t_a, a = 1, \ldots, L\}$ for a particular test utterance of accent a, compared against all the L target accent models, scores are further post-processed as,

$$t'_a = \log\frac{\exp(t_a)}{\frac{1}{L-1}\sum_{k\neq a}\exp(t_k)}, \tag{6}$$

where t'_a is the detection log-likelihood ratio, for a particular test utterance of accent a, scored against all the L target accent models.

III. Experimental Setup

A. Baseline System

To compare the attribute system recognition performance, two baseline systems were built. Both systems were trained using 56 dimensional SDC (49)-MFCC (7) feature vectors and they use the same UBM of 512 Gaussians. The first system is based on the conventional GMM-UBM system with adaptation similar to [16]. It uses 1 iteration to adapt the UBM to each target model. Adaptation consists of updating only the GMM mean vectors. The detection scores are then generated using a fast scoring scheme described in [40] using top 5 Gaussians. The second system uses i-vectors approach to classify accents. The i-vectors are of dimensionality 1000 and HLDA projected i-vectors of dimensionality 180.

B. Corpora

The "stories" part of the OGI Multi-language telephone speech corpus [41] was used to train the attribute detectors. This corpus has phonetic transcriptions for six languages: English, German, Hindi, Japanese, Mandarin, and Spanish. Data from each language were pooled together to obtain 5.57 hours of training and 0.52 hours of validation data.

A series of foreign accent recognition experiments were performed on the *FSD* corpus [42] which was developed to assess Finnish language proficiency among adults of different nationalities. We selected the oral responses portion of the exam, corresponding to 18 foreign accents. Since the number of utterances is small, 8 accents — Russian, Albanian, Arabic, English, Estonian, Kurdish, Spanish, and Turkish — with enough available data were used. The unused accents are, however, used in training UBM and the T-matrix. Each accent set is randomly split into a test and a train set. The test set consists of (approximately) 30% of the utterances, while the training set consists of the remaining 70% to train foreign accent recognizers in the FSD task. The raw audio files were partitioned into 30 sec chunks and re-sampled to 8 kHz. Statistics of the test and train portions are shown in Table II.

The NIST 2008 SRE corpus was chosen for the experiments on English foreign accent detection. The corpus has a rich metadata from the participants, including their age, language and smoking habits. It contains many L2 speakers whose native language is not English. Since the number of utterances in some foreign accents is small, 7 accents — Hindi (HIN), Thai (THA), Japanese (JPN), Russian (RUS), Vietnamese (VIE), Korean (KOR) and Chinese Cantonese (YUH) — with enough available utterances were chosen in this study. These accents are from the short2, short3 and 10sec portions, of the NIST 2008 SRE corpus. We used over 5000 utterances to train the UBM and total variability subspace in the NIST 2008 task. Table III shows the distribution of train and test portions in the English utterances. Speakers do not overlap between training and testing utterances both in the FSD and NIST corpora.

C. Attribute Detector Design

One-hidden-layer feed forward multi-layer perceptrons (MLPs) were used to implement each attribute detector. The

TABLE II: Train and test files distributions in each target accent in the FSD corpus. Duration is reported for only active speech frames.

Accent	#train files (hh:mm)	#test files	#speakers
Spanish	47 (00:26)	25	15
Albanian	60 (00:32)	29	19
Kurdish	61 (00:37)	32	21
Turkish	66 (00:39)	34	22
English	70 (00:37)	36	23
Estonian	122 (01:07)	62	38
Arabic	128 (01:15)	66	42
Russian	556 (03:15)	211	235
Total	1149 (08:46)	495	415

TABLE III: Train and test file distributions in the NIST 2008 SRE corpus. Duration is reported for only active speech frames.

Accent	#train files (hh:mm)	#test files	#speakers
Hindi	80 (03:39)	109	53
Russian	74 (03:32)	84	42
Korean	91 (03:05)	99	41
Japanese	53 (02:02)	73	41
Thai	70 (02:53)	93	52
Cantonese	68 (03:14)	92	50
Vietnamese	127 (04:01)	149	69
Total	563 (22:44)	699	348

number of hidden nodes with a sigmoidal activation function is 500. MLPs were trained to estimate attribute posteriors, and the training data were separated into "feature present", "feature absent", and "other" regions for every phonetic class used in this work. The classical back-propagation algorithm with a cross-entropy cost function was adopted to estimate the MLP parameters. To avoid over-fitting, the reduction in classification error on the development set was adopted as the stopping criterion. The attribute detectors employed in this study were actually just those used in [29].

Data-driven detectors are used to spot speech cues embedded in the speech signal. An attribute detector converts an input utterance into a time series that describes the level of presence (or level of activity) of a particular property of an attribute over time. A bank of 15 detectors (6 manner and 9 place) is used in this work, each detector being individually designed to spot of a particular event. Each detector is realized with three single hidden layer feed-forward ANNs (artificial neural networks) organized in a hierarchical structure and trained on sub-band energy trajectories extracted through 15-band mel-frequency filterbank. For each critical band, a window of $310ms$ centered around the frame being processed is considered and split in two halves: left-context and right-context [43]. Two independent front-end ANNs ("lower nets") are trained on those two halves to generate, left- and right-context speech attribute posterior probabilities. The outputs of the two lower nets are then sent to the third ANN that acts as a merger and gives the attribute-state posterior probability of the target speech attribute.

D. Evaluation Metrics

System performance is reported in terms of *equal error rate* (EER) and average detection cost (C_{avg}) [44]. Results are reported per each accent for a cosine scoring classifier. C_{avg} is defined as [44],

$$C_{\text{avg}} = \frac{1}{M}\sum_{j=1}^{M} C_{\text{DET}}(L_j), \tag{7}$$

where $C_{\text{DET}}(L_j)$ is the detection cost for subset of test segments trials for which the target accent is L_j and M is the number of target languages. The per target accent cost is then,

$$\begin{aligned} C_{\text{DET}}(L_j) &= C_{\text{miss}} P_{\text{tar}} P_{\text{miss}}(L_a) \\ &+ C_{\text{fa}}(1-P_{\text{tar}})\frac{1}{J-1}\sum_{k\neq j} P_{\text{fa}}(L_j, L_k). \end{aligned} \tag{8}$$

The miss probability (or false rejection rate) is denoted by P_{miss}, i.e., a test segment of accent L_i is rejected as being in that accent. On the other hand $P_{\text{fa}}(L_i, L_k)$ denotes the probability when a test segment of accent L_k is accepted as being in accent L_i. It is computed for each target/non-target accent pairs. The costs, C_{miss} and C_{fa} are both set to 1 and P_{tar}, the prior probability of a target accent, is set to 0.5 following [44].

IV. Results

A. Accent Recognition Performance on the FSD corpus

Table IV reports foreign accent recognition results for several systems on the FSD corpus. The results in the first two rows indicate that i-vector modeling outperforms the GMM-UBM technique when the same input features are used, which is in line with findings in [10], [45]. The results in the last two rows, in turn, indicate that the i-vector approach can be further enhanced by replacing spectral vectors with attribute features. In particular, the best performance is obtained using manner attribute features within the i-vector technique, yielding a C_{avg} of 5.80, which represents relative improvements of 45% and 15% over the GMM-UBM and the conventional i-vector approach with SDC+MFCC features, respectively. The FSD task is quite small, which might make the improvements obtained with the attribute system not statistically different from those delivered by the spectral-based system. We therefore decided to run a proper statistical significance test using a dependent Z-test according to [46]. We applied the statistical test for comparing per target accents EERs between attribute systems and SDC-MFCC i-vector system. In Table V, we indicated in boldface cases where the proposed attribute-based foreign accent recognition techniques outperform the spectral-based one. To exemplify, Z-test results in the second column of Table V demonstrates that the manner system significantly outperforms the SDC-MFCC i-vector system on 7 out of 8 accents. For the sake of completeness, we have also compared manner and place of articulation systems, and we have reported the Z-test results in the third column of Table V.

To verify that we are not recognizing the channel variability, we followed the procedure highlighted in [47], where

TABLE IV: Baseline and attribute systems results in terms of $\mathrm{EER_{avg}}$ and $\mathrm{C_{avg}}$ in the FSD corpus. In parentheses, the final dimensionality of the feature vectors sent to the back-end. In manner system, for 7 out of 8 accents, the difference between EERs is significant at a confidence level of 95% if $Z \geq 1.960$.

Feature (dimensionality)	Classifier	$\mathrm{EER_{avg}}$(%)	$\mathrm{C_{avg}} \times 100$
SDC+MFCC (56)	GMM-UBM	19.03	10.56
SDC+MFCC (56)	i-vector	12.60	6.85
Place (27)	i-vector	10.37	6.00
Manner (18)	i-vector	**9.21**	**5.80**

TABLE V: In the first two columns, the Z-test results per target accent EERs at the EER threshold between the proposed attribute- and spectral-based system performance on the FSD corpus are reported. The difference between EERs is significant at a confidence level of 95% if $Z \geq 1.960$. Boldface values refer to cases in which our solution significantly outperforms the SDC-MFCC system. The third column shows the same Z-test results between manner- and place-based systems, where manner is significantly better than place if the score is in boldface.

Accents	**Place/SDC-MFCC**	**Manner/SDC-MFCC**	**Manner/Place**
Albanian	1.1041	1.6503	**2.2866**
Arabic	**5.9139**	**5.6975**	1.0587
English	**1.9973**	**4.3714**	**3.0224**
Estonian	0.4907	**2.2240**	1.2108
Kurdish	**5.1326**	**3.1453**	**2.2361**
Russian	**2.3955**	**5.2633**	**3.1523**
Spanish	**5.4506**	**2.2105**	**2.3521**
Turkish	**4.9694**	**1.9604**	**3.6600**

the authors performed language recognition experiments on speech and non-speech frames separately. The goal of the authors was to demonstrate that if the system performance on the non-speech frames is comparable with that attained using speech frames, then the system is actually modeling the channel and not language variability. Therefore, we have first split data into speech and non-speech frames. Then we have computed the $\mathrm{EER_{avg}}$ on the non-speech frames, which was equal to 40.51% and 40.18% in manner and place cases, respectively. The $\mathrm{EER_{avg}}$ on the speech frames was instead equal to 8.48% and 14.20% in the manner and place systems, respectively. These results suggest that our technique is not modeling channel effects.

Next we explore different architectural configurations to assess their effect on the recognition accuracy.

1) Effect of i-vector dimensionality on the FSD corpus: In Table IV, we showed that attribute system outperforms the baseline spectral system in foreign accent recognition. Here, we turn our attention to the choice of i-vector dimensionality used to train and evaluate different models. Figure 4 shows recognition error rates on the FSD corpus as a function of i-vector size. Results indicate that better system performance can be attained by increasing the i-vector dimensionality up to 1000, which is inline with the findings reported in [22]. However, further increasing the i-vector dimensionality to 1200, or 1400 degraded the recognition accuracy. For example, $\mathrm{C_{avg}}$ increased to 6.10 and 6.60 from the initial 5.80 for the manner-based foreign accent recognition system with i-vector

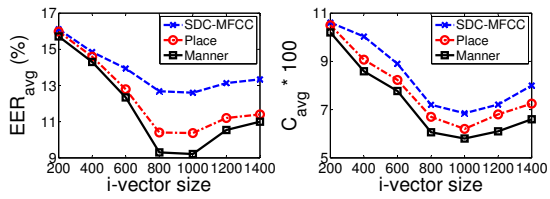


Fig. 4: Recognition error rates as a function of i-vector dimensionality on the FSD corpus.

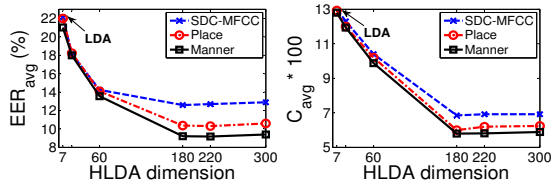


Fig. 5: Recognition error rates as a function of HLDA dimension on the FSD corpus. I-vectors are of dimensionality 1000. For lower HLDA dimensions, i.e., 7, 20 and 60, the systems attain lower recognition accuracies.

dimensionality of 1200 and 1400, respectively.

We also investigated the effect of HLDA dimensionality reduction algorithm on recognition error rates using 6 different HLDA output dimensionalities on the FSD corpus. Figure 5 shows that the optimal HLDA dimension is around 180, yielding C_{avg} of 5.8 and 6 in the manner and place systems, respectively. For lower HLDA dimensions, i.e., 7, 20 and 60, the systems attain lower recognition accuracies as shown. Comparing HLDA results in Figure 5 with LDA, the recognition error rates increase to $\mathrm{EER_{avg}}$ of 21.65% and 21.87% in manner and place systems, respectively. The output dimensionality of LDA is then restricted to maximum of seven.

2) Effect of training set size and testing utterance length on the FSD corpus: To demonstrate the recognition error rates as a function of training set size in this study, we split the Finnish training i-vectors into portions of 20%, 40%, 60%, 80% and 100% of the whole training i-vectors within each model in such a way that each individual portion contains the data from previous portion. Fixing the amount of test data, we experimented with each training data portion to report the recognition error rates as a function of training data size. Results in Figure 6 shows that the proposed attribute-based foreign accent recognition system outperforms the spectral-based system in all the cases (i.e., independently of the amount of training data). Further to see the effect of test data length on recognition error rates, we extracted new i-vectors from the 20%, 40%, 60%, 80% and 100% of *active speech frames* and used them in evaluation. Results in Figure 7, which refers to the FSD corpus, indicate that the proposed attribute-based accent recognition system compares favorably to the SDC-MFCC system in all the cases.

3) Effect of Temporal Context – FSD corpus: In Section II-B, it was argued that temporal information may be beneficial

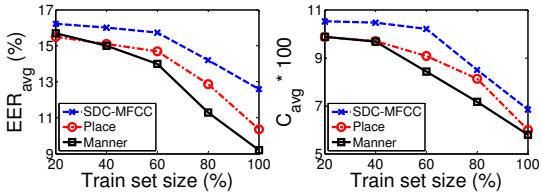


Fig. 6: Recognition error rates as a function of training set size on the FSD corpus. Increasing training set size within each target accent models degrades recognition error rates.

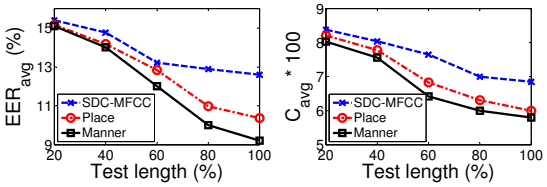


Fig. 7: Recognition error rates as a function of testing utterance length on the FSD corpus. Different portions of active speech segments were used to extract evaluation i-vectors.

to accent recognition. Figure 4 indicates that $\mathrm{C_{avg}}$ attains minima at context sizes 10 and 20 frames, for the place and manner features, respectively. Optimum for the PCA-combined features occurs at 10 frames. Increasing the context size beyond 20 frames negatively affects recognition accuracy for all the evaluated configurations. In fact, we tested context window spanning up to 40 adjacent frames, but that caused numerical problems during UBM training, leading to singular covariance matrices. Hence, context size in the range of 10 to 20 frames appears a suitable trade-off between capturing contextual information while retaining feature dimensionality manageable for our classifier back-end.

Table VI shows results for several configurations of the proposed technique and optimal context window sizes selected according to Figure 8. Systems using context dependent information are indicated by adding the letters CD in front of their name. The last two rows show the result for context-independent attribute systems for reference purposes. Table VI demonstrates that context information is beneficial for foreign accent recognition. The best performance is obtained by concatenating C=20 adjacent manner feature frames followed by PCA to reduce the final vector dimensionality to d=48. A 14% relative improvement, in terms of $\mathrm{C_{avg}}$, over the context-independent manner system (last row) is obtained by adding context information.

4) Effect of Feature Concatenation on the FSD corpus: We now turn our attention to the effects of feature concatenation on the accent recognition performance. The first row of Table VII shows that $\mathrm{C_{avg}}$ of 5.70 is obtained by appending the place features with the SDC+MFCC features, which yields a relative improvement of 5% over the place system (third last row). A 12% relative improvement over the manner system (second last row) is obtained by concatenating the

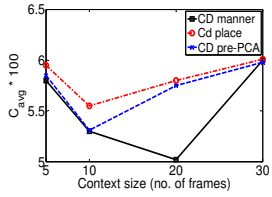


Fig. 8: $\mathrm{C_{avg}}$ as a function of the context window size on the FSD corpus. Context dependent (CD) manner and place features attain the minimum $\mathrm{C_{avg}}$ at context sizes 10 and 20 frames, respectively. In pre-PCA, PCA is applied to combined manner and place vectors.

TABLE VI: Recognition results for several attribute systems and different context window sizes. C represents the length of context window, and d the vector dimension after PCA. PCA can be applied either before (pre-PCA) or after (post-PCA) concatenating manner and place vectors.

System (context, dimension)	$\mathrm{EER_{avg}}$(%)	$\mathrm{C_{avg}} \times 100$
CD Place (C=10, d=31)	8.87	5.55
CD post-PCA (C=10, d=70)	8.20	5.43
CD pre-PCA (C=10, d=60)	7.97	5.31
CD Manner (C=20, d=48)	**7.38**	**5.02**
Place (27)	10.37	6.00
Manner (18)	9.21	5.80

SDC+MFCC features and the manner features, yielding $\mathrm{C_{avg}}$ of 5.13 (the second row). If context-dependent information is used before forming the manner-based vector to be concatenated with the SDC+MFCC features, a further improvement is obtained, as the third row of Table VII indicates. Specifically, $\mathrm{C_{avg}}$ of 4.74 is obtained by using a context of 20 frames followed by PCA reduction down to 48 dimensions (C=20, d=48). The result represents 19% relative improvement over the use of CD manner-only score with the same context window and final dimensionality (last row).

For the sake of completeness, Table VII shows also results obtained by concatenating manner and place attributes, which is referred to as *Manner+Place* system. This system obtains $\mathrm{C_{avg}}$ of 5.51, which represents 5% and 8% relative improvements over the basic manner and place systems, respectively. In contrast, no improvement is obtained by concatenating context-dependent manner and place systems (see the row labeled CD Manner (C=20, d=48) + CD Place (C=10, d=31)) over context-dependent manner system (last row).

5) Detection Performance versus Target Language – FSD corpus: Table VIII shows language-wise results on the FSD task. The so-called leave-one-speaker-out (LOSO) technique, already used in [10], was adopted to generate these results and to compensate for lack of sufficient data in training and evaluation. For every target accent, each speaker's utterances are left out one at a time while the remaining utterances are used in training the corresponding accent recognizer. The held-out utterances are then used as the evaluation utterances.

The CD manner-based accent recognition system was selected for this experiment, since it outperformed the place-

TABLE VII: Results on the FSD corpus after feature concatenation (+). In parentheses, the final dimension of the feature vectors sent to the back-end.

System (feature dimensionality)	Performance	
(SDC+MFCC) Vector + Attribute Vector	$\mathrm{EER_{avg}}(\%)$	$\mathrm{C_{avg}} \times 100$
(SDC+MFCC) + Place (83)	9.14	5.70
(SDC+MFCC) + Manner (74)	7.78	5.13
(SDC+MFCC) + CD Manner (C=20, d=48) (104)	**6.18**	**4.74**
Feature Concatenation (+) within Attributes	$\mathrm{EER_{avg}}(\%)$	$\mathrm{C_{avg}} \times 100$
Manner + Place (45)	8.34	5.51
CD Manner (C=20, d=48) + CD Place (C=10, d=31) (79)	8.00	5.34
Basic Accent Recognition System	$\mathrm{EER_{avg}}(\%)$	$\mathrm{C_{avg}} \times 100$
SDC+MFCC (56)	12.60	6.85
Place (27)	10.37	6.00
Manner (18)	9.21	5.80
CD Manner (C=20, d=48) (48)	7.38	5.02

TABLE VIII: *Per language results in terms of EER % and* $\mathrm{C_{DET}} \times 100$ *on the FSD corpus. Results are reported for the CD Manner (C=20, d=48).*

Accents	EER %	$\mathrm{C_{DET}} \times 100$
English	15.11	7.00
Estonian	14.54	6.33
Russian	13.08	6.30
Kurdish	13.00	6.11
Arabic	12.55	6.10
Albanian	11.43	6.07
Spanish	10.74	5.75
Turkish	8.36	5.52
Total (average)	12.35	6.14

based one. Furthermore, since we have already observed that the performance improvement obtained by combining manner- and placed-based information is not compelling, it is preferable to use a less complex system.

Table VIII indicates that Turkish is the easiest accent to detect. In contrast, English and Estonian are the hardest accents to detect. Furthermore, languages with different subfamily from Finnish, are among the easiest to deal with. Nonetheless, the last row of Table VIII shows an $\mathrm{EER_{avg}}$ and a $\mathrm{C_{avg}}$ higher than the corresponding values reported in Table VI. This might be explained recalling that the unused accents employed to train UBM, T-matrix and the HLDA in LOSO induces a mismatch between model training data and the hyper-parameter training data which degrades the recognition accuracy [10].

It is interesting to study the results of Table VIII a bit deeper to understand which language pairs are easier to confuse. Here we treat the problem as foreign accent identification task. Table IX shows the confusion matrix. The diagonal entries demonstrate that correct recognition is highly likely. Taking Turkish as the language with highest recognition accuracy, out of 30 misclassified Turkish test segments, 10 are classified as Arabic. That seems to be a reasonable result, since Turkey is bordered by two Arabic countries, namely Syria and Iraq. In addition, Turkish shares common linguistic features with Arabic. With respect to Albanian as one of the languages in the middle: 11 out of 26 misclassified test segment are assigned to the Russian class. That might be explained considering

TABLE IX: Confusion matrix on the Finnish accent recognition task. Results are reported for the CD manner (C=20, d=48).

		Predicted label							
		TUR	SPA	ALB	ARA	KUR	RUS	EST	ENG
True label	TUR	**70**	3	1	10	5	5	2	4
	SPA	1	**51**	3	8	2	2	3	2
	ALB	1	3	**62**	3	1	11	5	2
	ARA	12	9	7	**128**	10	9	8	8
	KUR	9	3	3	6	**60**	5	3	4
	RUS	43	30	51	20	16	**379**	25	26
	EST	6	8	8	12	6	13	**120**	13
	ENG	7	10	3	6	3	7	6	**63**

TABLE X: English results in terms of $\mathrm{EER_{avg}}(\%)$ and $\mathrm{C_{avg}}$ on the NIST 2008 corpus. In parentheses, the final dimensionality of the feature vectors sent to the back-end.

Feature (dimensionality)	Classifier	$\mathrm{EER_{avg}}(\%)$	$\mathrm{C_{avg}} \times 100$
SDC+MFCC (56)	GMM-UBM	16.94	9.00
SDC+MFCC (56)	i-vector	13.82	7.87
Place (27)	i-vector	12.00	7.27
Manner (18)	i-vector	11.09	6.70
CD Manner (C=20, d=48)	i-vector	**10.18**	**6.30**

that Russian has a considerable influence on the Albanian vocabulary. Russian is one of the most difficult languages to detect, and 43 samples are wrongly recognized as Turkish. The latter outcome can be explained recalling that Russian has some words with Turkish roots; moreover, the two languages have some similarities in terms of pronunciation.

B. Results on the NIST 2008 corpus

Up to this point, we have focused on the FSD corpus to optimize parameters. These parameters are: the UBM and i-vector size, the HLDA dimensionality, and the context window size. The first three parameters, i.e. UBM size 512, i-vector dimensionality 1000 and HLDA dimensionality 180 were optimized in [10] while the context window was set to $C = 20$ for manner attributes based on our analysis in the present study. We now use the optimized values to carry out experiments on English data.

Table X compares results of the proposed and baseline systems on the NIST 2008 SRE corpus. As above, manner- and place-based systems outperform the SDC+MFCC-based i-vector system, yielding 15% and 8% relative improvements in $\mathrm{C_{avg}}$, respectively. These relative improvements are lower compared to the corresponding results for Finnish, which is understandable considering that the parameters were optimized on the FSD data. The best recognition results are obtained using a context window of C=20 adjacent frames and dimensionality reduction to d=48 features via PCA. Similar to FSD task, different architectural alternatives are now investigated to further boost system performance.

1) Effect of Feature Concatenation on the NIST 2008 corpus: Feature concatenation results on the NIST 2008 task are shown in Table XI. Similar to findings on FSD, accuracy is enhanced by combining SDC+MFCC and attribute features. The largest relative improvement is obtained by combining SDC+MFCC and CD manner features (third row in Table

TABLE XI: Results on the NIST 2008 corpus after feature concatenation (+). In parentheses, the final dimensionality of the feature vectors sent to the back-end.

System (feature dimensionality)	**Performance**	
(SDC+MFCC) Vector + Attribute Vector	$\mathrm{EER_{avg}}$(%)	$\mathrm{C_{avg}} \times 100$
(SDC+MFCC)+Place (83)	11.20	6.82
(SDC+MFCC)+Manner (74)	10.01	6.24
(SDC+MFCC)+CD Manner (C=20, d=48) (104)	**8.56**	**5.73**
Feature Concatenation (+) within Attributes	$\mathrm{EER_{avg}}$(%)	$\mathrm{C_{avg}} \times 100$
Manner+Place	10.50	6.40
Basic Accent Recognition system	$\mathrm{EER_{avg}}$(%)	$\mathrm{C_{avg}} \times 100$
SDC+MFCC (56)	13.82	7.87
Place (27)	12.00	7.27
Manner (18)	11.09	6.70
CD Manner (C=20, d=48)	10.18	6.30

TABLE XII: Per-language results in terms of EER % and $\mathrm{C_{DET}} \times 100$ for the i-vector system in the NIST 2008 corpus. Results are reported for CD manner (C=20, d=48)

Accents	EER %	$\mathrm{C_{DET}} \times 100$
Cantonese	16.48	8.46
Hindi	14.97	7.91
Vietnamese	14.04	7.30
Russian	12.09	7.57
Korean	11.54	6.96
Japanese	10.84	6.62
Thai	10.59	6.35
Total (average)	12.93	7.31

XI), yielding $\mathrm{C_{avg}}$ of 5.73. As for FSD, improvement is also obtained by concatenating manner and place features, with final $\mathrm{C_{avg}}$ of 6.40, which represents 7% relative improvement over the basic configurations in the second and third last rows. Nonetheless, higher accuracy is obtained by the CD manner system, shown in the last row.

2) Detection Performance versus Target Language – NIST 2008 corpus: Table XII shows per-accent detection accuracy on the NIST 2008 task. Similar to the FSD experiments, the LOSO technique is applied to make better use of the limited training and testing data. Cantonese attains the lowest recognition accuracy with $\mathrm{C_{DET}}$ of 8.46; and the easiest accent is Thai with $\mathrm{C_{DET}}$ of 6.35. The confusion matrix is shown in Table XIII. It is obvious that East Asian languages, such as Korean, Japanese, Vietnamese and Thai are frequently confused with Cantonese. For example, Thai is the easiest accent to detect, yet 15 out of the 37 misclassified test segments were classified as Cantonese. Thai and Cantonese are both from the same Sino-Tibetan language family; therefore, these languages share similar sound elements. Furthermore, the same set of numbers from one to ten is used for both languages.

Russian and Hindi are both from the Indo-European language group. Hence these languages have many words and phrases in common. These similarities might explain why 12 out of 36 misclassified Russian segments were classified as Hindi. Similarly, 14 out of 48 misclassified Hindi segments were assigned to the Russian language.

TABLE XIII: Confusion matrix of the English results corresponding to Table XII. Results are reported for CD manner (C=20, d=48)

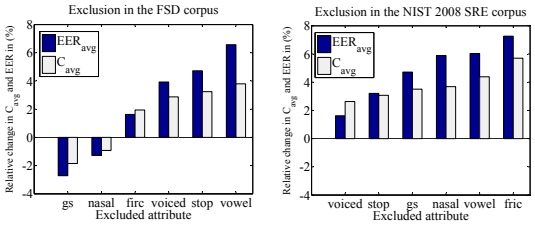
		Predicted label						
		THA	JPN	KOR	RUS	VIE	HIN	CAN
True label	THA	**126**	3	4	3	4	8	15
	JPN	3	**98**	4	2	7	2	10
	KOR	3	5	**145**	6	7	5	17
	RUS	4	3	3	**120**	4	12	10
	VIE	10	16	6	6	**200**	4	33
	HIN	4	4	6	14	5	**128**	15
	CAN	15	10	11	6	14	6	**96**

C. Effect of Individual Attribute on Detection Performance

We now investigate the relative importance of each individual manner attribute and the voiced attribute on both FSD and NIST 2008. We selected manner-based system as it outperformed place-based system both in both FSD and NIST 2008 (Tables IV and X). A 15-dimensional feature vector is formed by leaving out one of these attributes one at a time. The full i-vector system is then trained from scratch using the feature vectors without the excluded attribute. By comparing the change in $\mathrm{EER_{avg}}$ and $\mathrm{C_{avg}}$ of such system relative to the system utilizing all the 15 features allows us to quantify the relative importance of that attribute. When no context information is used, $\mathrm{EER_{avg}}$ and $\mathrm{C_{avg}}$ are 9.21% and 5.80, respectively.

Figure 9a reveals that excluding vowel, stop, voiced, or fricative attributes increases both $\mathrm{C_{avg}}$ and $\mathrm{EER_{avg}}$, indicating the importance of these attributes. In contrast, nasal and glide are not individually beneficial, since both $\mathrm{C_{avg}}$ and $\mathrm{EER_{avg}}$ show a negative relative change. Finnish has a very large vowel space (with 8 vowels) including vowel lengthening. Non-native Finnish speakers may thus have troubles when trying to produce vowels in a proper way, and that shows the L1 influence. This may explain why vowels are individually useful in foreign accent recognition for Finnish.

Figure 9b shows that *all* speech attributes are individually useful in detecting L2 in an English spoken sentence. We recall that $\mathrm{EER_{avg}}$ and $\mathrm{C_{avg}}$ are 11.09% and 6.70, respectively, when no context information is used. Hence, leaving out any of these attributes from the final feature vector, increases the error rates. Fricative and vowel are individually most important, while, voiced and stop attributes are less important. It is known that pronouncing English fricatives is difficult for some L2 speakers [48], [49]. For example, some Russian speakers pronounce dental fricatives /ð/ and /θ/ as /t/ and /d/, respectively [50]. With respect to the vowel class, some East Asian speakers find it difficult to pronounce English vowels, thus producing L1 influence. For example, English contains more vowel sounds than Chinese languages [51]. This may cause Chinese learners of English to have difficulties with pronunciation. Koreans may also have also difficulty pronouncing the sound /ɔ/ which does not exist in Korean language and is frequently substituted with the sound /o/ in Korean [52].



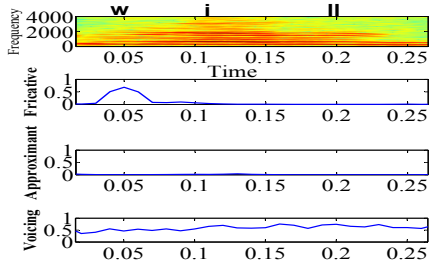
(a) FSD task (b) NIST 2008 task

Fig. 9: Exclusion experiment: relative change in the error rates as one attribute is left out. Positive relative change indicates increment in the error rates.

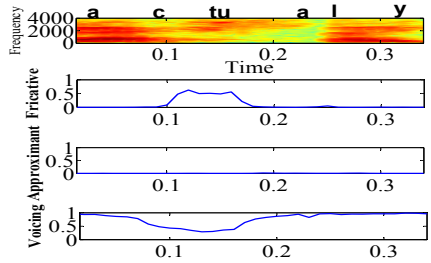
D. Diagnostic Information of Attribute Features

Besides improving the accuracy of state-of-the-art automatic foreign accent recognizer, the proposed technique provides a great deal of diagnostic information to pinpoint *why* it works well in one instance and then fail badly in another. To exemplify, Figure 10 shows analysis of two different spoken words uttered by native Russian and Cantonese speakers in the NIST 2008 SRE corpus on which the proposed attribute-based technique was successful, but the spectral-based SDC+MFCC technique failed. Figure 10a shows the spectrogram along with fricative and the approximant detection curves for the word "will" uttered by a native Russian speaker. Although /w/ belongs to the approximant class, the corresponding detection curve is completely flat. In contrast, a high level of activity is seen in the fricative detector. This can be explained noting that Russian does not have the consonant /w/, and Russian speakers typically substitute it with /v/ [53], which is a fricative consonant. Figure 10b, in turn, signifies that consonant sounds, except nasals and semivowels, are all voiceless in Cantonese [54]. Although /c/ (pronounced as a /k/) and /tu/ (pronounced as a /tʃ/) are voiced consonants in English, voicing activity is less pronounced in the time frame spanning the /c/ and /tu/ consonants, which is a specific feature of Cantonese speakers [54].

Incidentally, such information could also be useful in computer-assisted language learning system to detect mispronunciations and give some proper feedback to the user.



(a) Native Russian speaker substitutes approximant /w/ with fricative /v/.



(b) Consonants in Cantonese are all voiceless.

Fig. 10: The informative nature of the proposed accent recognition system for two spoken utterances from native Russian and Cantonese speakers. For these utterances, attribute-based technique has been successful but the spectral-based technique has failed.

V. Conclusion

In this paper, an automatic foreign language recognition system based on universal acoustic characterization has been presented.

Taking inspiration from [30], the key idea is to describe any spoken language with a common set of fundamental units that can be defined "universally" across all spoken languages. Phonetic features, such as manner and place of articulation, are chosen to form this unit inventory and used to build a set of language-universal attribute models with data-driven modeling techniques.

The proposed approach aims to unify within a single framework phonotactic and spectral based approach to automatic foreign accent recognition. The leading idea is to take the advantages of the subspace modeling techniques without discharging the valuable information provided by the phonotactic-based methods. To this end, a spoken utterance is processed through a set of speech attribute detectors in order to generate attribute-based feature streams representing foreign accent cues. These feature streams are then modeled within the state-of-the-art i-vector framework.

Experimental evidence on two different foreign accent recognition tasks, namely Finnish (FSD corpus) and English (NIST 2008 corpus), has demonstrated the effectiveness of the proposed solution, which compares favourably with state-of-the-art spectra-based approaches. The proposed system based on manner of articulation has achieved a relative improvement of 45% and 15% over the conventional GMM-UBM and the i-vector approach with SDC+MFCC vectors, respectively, on the FSD corpus. The place-based system has also outperformed the SDC+MFCC-based i-vector system with a 8% C_{avg} relative improvement. The difficulty at robust modeling of place of articulation causes that smaller relative improvement. It was also noticed that context information improves system performance.

We plan to investigate how to improve the base detector accuracy of place of articulation. In addition, we will investigate phonotactic [55] and deep learning language recognition systems [56] in the foreign accent recognition task. Especially, we are interested to find out whether in terms of classifier

fusion complementary information exist in those systems and our proposed method.

REFERENCES

[1] J. H. Hansen and L. M. Arslan, "Foreign accent classification using source generator based prosodic features," in *Proc. of ICASSP*, 1995, pp. 836–839.

[2] V. Gupta and P. Mermelstein, "Effect of speaker accent on the performance of a speaker-independent, isolated word recognizer," *J. Acoust. Soc. Amer.*, vol. 71, no. 1, pp. 1581–1587, 1982.

[3] R. Goronzy, S. Rapp, and R. Kompe, "Generating non-native pronunciation variants for lexicon adaptation," *Speech Communication*, vol. 42, no. 1, pp. 109–123, 2004.

[4] L. M. Arslan and J. H. Hansen, "Language accent classification in American English," *Speech Communication*, vol. 18, no. 4, pp. 353–367, 1996.

[5] P. Angkititraku and J. H. Hansen, "Advances in phone-based modeling for automatic accent classification," in *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, 2006, pp. 634–646.

[6] GAO, *Border Security: Fraud Risks Complicate States Ability to Manage Diversity Visa Program*. DIANE Publishing, 2007. [Online]. Available: http://books.google.com/books?id=PfmuLdR66qwC

[7] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," Ph.D. dissertation, Columbia University, 2011.

[8] J. Nerbonne, "Linguistic variation and computation (invited talk)," in *Proc. of EACL*, 2003, pp. 3–10.

[9] J. Flege, C. Schirru, and I. MacKay, "Interaction between the native and second language phonetic subsystems," *Speech Communication*, vol. 40, no. 4, pp. 467–491, 2003.

[10] H. Behravan, V. Hautamäki, and T. Kinnunen, "Factors affecting i-vector based foreign accent recognition: a case study in spoken Finnish," *Speech Communication*, vol. 66, pp. 118–129, 2015.

[11] Asher, J. J., and R. Garcia, "The optimal age to learn a foreign language," *Modern languages*, vol. 38, pp. 334–341, 1969.

[12] M. Zissman, T. Gleason, D. Rekart, and B. Losiewicz, "Automatic dialect identification of extemporaneous conversational latin American Spanish speech," in *Proc. of ICASSP*, 1995, pp. 777–780.

[13] W. M. Campbell, J. P. Campbell, and D. A. Reynolds, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20 (2-3), no. 2-3, pp. 210–229, 2005.

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

[15] 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 *Proc. of ICSLP*, 2002, pp. 89–92.

[16] P. A. Torres-Carrasquillo, T. P. Gleason, and D. A. Reynolds, "Dialect identification using Gaussian mixture models," in *Proc. of Odyssey*, 2004, pp. 757–760.

[17] G. Liu and J. H. Hansen, "A systematic strategy for robust automatic dialect identification," in *Proc. of EUSIPCO*, 2011, pp. 2138–2141.

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

[19] C. Teixeira, I. Trancoso, and A. J. Serralheiro, "Recognition of non-native accents," in *Proc. of EUROSPEECH*, 1997, pp. 2375–2378.

[20] K. Kumpf and R. W. King, "Automatic accent classification of foreign accented Australian English speech," in *Proc. of ICSLP*, 1996, pp. 1740–1742.

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

[22] H. Behravan, V. Hautamäki, and T. Kinnunen, "Foreign accent detection from spoken Finnish using i-vectors," in *Proc. of INTERSPEECH*, 2013, pp. 79–83.

[23] A. DeMarco and S. J. Cox, "Iterative classification of regional British accents in i-vector space," in *Proc. of SIGML*, 2012, pp. 1–4.

[24] N. F. Chen, W. Shen, and J. P. Campbell, "A linguistically-informative approach to dialect recognition using dialect-discriminating context-dependent phonetic models," in *Proc. of ICASSP*, 2010, pp. 5014–5017.

[25] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *Proc. of INTERSPEECH*, 2004, pp. 109–112.

[26] S. M. Siniscalchi and C.-H. Lee, "A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition," *Speech Communication*, vol. 51, pp. 1139–1153, 2009.

[27] C.-H. Lee and S. M. Siniscalchi, "An information-extraction approach to speech processing: Analysis, detection, verification, and recognition," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1089–1115, 2013.

[28] M.-J. Kolly and V. Dellwo, "Cues to linguistic origin: The contribution of speech temporal information to foreign accent recognition," *Journal of Phonetics*, vol. 42, no. 1, pp. 12–23, 2014.

[29] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target specific training data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 875–887, 2012.

[30] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Universal attribute characterization of spoken languages for automatic spoken language recognition," *Computer Speech & Language*, vol. 27, no. 1, pp. 209–227, 2013.

[31] H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C.-H. Lee, "Introducing attribute features to foreign accent recognition," in *Proc. of ICASSP*, 2014, pp. 5332–5336.

[32] M. Loog and R. P. Duin, "Linear dimensionality reduction via a heteroscedastic extension of LDA: The Chernoff criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 732–739, 2004.

[33] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 788–798, 2011.

[34] M. Diez, A. Varona, M. Penagarikano, L. J. Rodriguez-Fuentes, and G. Bordel, "Dimensionality reduction of phone log-likelihood ratio features for spoken language recognition," in *Proc. of INTERSPEECH*, 2013, pp. 64–68.

[35] M. Díez, A. Varona, M. Peñagarikano, L. J. Rodríguez-Fuentes, and G. Bordel, "New insight into the use of phone log-likelihood ratios as features for language recognition," in *Proc. of INTERSPEECH*, 2014, pp. 1841–1845.

[36] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Phonotactic language recognition using MLP features," in *Proc. of INTERSPEECH*, 2012.

[37] D. Matrouf, N. Scheffer, B. G. B. Fauve, and J.-F. Bonastre, "A straightforward and efficient implementation of the factor analysis model for speaker verification," in *Proc. of INTERSPEECH*, 2007, pp. 1242–1245.

[38] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[39] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of INTERSPEECH*, 2006, pp. 1471–1474.

[40] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[41] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika, "The OGI multi-language telephone speech corpus," in *Proc. of ICSLP*, 1992, pp. 895–898.

[42] "Finnish national foreign language certificate corpus," http://yki-korpus.jyu.fi.

[43] P. Schwarz, P. Matějaka, and J. Cernock, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. of ICASSP*, 2006, pp. 325–328.

[44] H. Li, K. A. Lee, and B. Ma, "Spoken language recognition: From fundamentals to practice," *Proceedigns of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[45] A. DeMarco and S. J. Cox, "Native accent classification via i-vectors and speaker compensation fusion," in *Proc. of INTERSPEECH*, 2013, pp. 1472–1476.

[46] S. Bengio and J. Mariéthoz, "A statistical significance test for person authentication," in *Proc. of Odyssey*, 2004, pp. 237–244.

[47] H. Bořil, A. Sangwan, and J. H. L. Hansen, "Arabic dialect identification - 'is the secret in the silence?' and other observations," in *Proc. of INTERSPEECH*, 2012, pp. 30–33.

[48] M. Timonen, "Pronunciation of the English fricatives: Problems faced by native Finnish speakers," Ph.D. dissertation, University of Iceland, 2011.

[49] L. Enli, "Pronunciation of English consonants, vowels and diphthongs of Mandarin-Chinese speakers," *Studies in Literature and Language*, vol. 8, no. 1, pp. 62–65, 2014.

[50] U. Weinreich, *Languages in Contact*. The Hague: Mouton, 1953.

[51] D. Deterding, "The pronunciation of English by speakers from China," *English World-Wide*, vol. 27, no. 2, pp. 157–198, 2006.
[52] B. Cho, "Issues concerning Korean learners of English: English education in Korea and some common difficulties of Korean students," *English World-Wide*, vol. 1, no. 2, pp. 31–36, 2004.
[53] I. Thompson, "Foreign accents revisited: The English pronunciation of Russian immigrants," *Language Learning*, vol. 41, no. 2, pp. 177–204, 1991.
[54] T. T. N. Hung, "Towards a phonology of Hong Kong English," *World Englishes*, vol. 19, no. 3, pp. 337–356, 2000.
[55] 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.
[56] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. of ICASSP*, 2014, pp. 5337–5341.

Hamid Behravan received the B.Sc. degree in Electrical Engineering from the Semnan University in 2010. He received the M.Sc. degree in Computer Science from the University of Eastern Finland in 2012. He is currently a Ph.D. student in Computer Science in the same university. From 2013 to 2015, he has worked as a project researcher for the University of Turku, funded by Kone Foundation. His research interests are in the area of speech processing, with current focus on automatic language and foreign accent recognition. In addition, he is interested in nonlinear analysis of speech signals.

Ville Hautamäki received the M.Sc. degree in Computer Science from the University of Joensuu (currently known as the University of Eastern Finland), Finland in 2005. He received the Ph.D. degree in Computer Science from the same university in 2008. He has worked as a research fellow at the Institute for Infocomm Research, A*STAR, Singapore. In addition, he has worked as a post-doctoral researcher in University of Eastern Finland, funded by Academy of Finland. Currently he is working as a senior researcher in the same university. His current research interests consists of recognition problems from speech signals, such as speaker recognition and language recognition. In addition, he is interested in application of machine learning to novel tasks.

Sabato Marco Siniscalchi is an Associate Professor at the University of Enna "Kore" and affiliated with the Georgia Institute of Technology. He received his Laurea and Doctorate degrees in Computer Engineering from the University of Palermo, Palermo, Italy, in 2001 and 2006, respectively. In 2006, he was a Post Doctoral Fellow at the Center for Signal and Image Processing (CSIP), Georgia Institute of Technology, Atlanta, under the guidance of Prof. C.-H. Lee. From 2007 to 2009, he joined the Norwegian University of Science and Technology, Trondheim, Norway, as a Research Scientist at the Department of Electronics and Telecommunications under the guidance of Prof. T. Svendsen. In 2010, he was a Researcher Scientist at the Department of Computer Engineering, University of Palermo, Italy. He acts as an an associate editor in IEEE/ACM Transactions on Audio, Speech and Language Processing. His main research interests are in speech processing, in particular automatic speech and speaker recognition, and language identification.

Tomi Kinnunen received the Ph.D. degree in computer science from the University of Eastern Finland (UEF, formerly Univ. of Joensuu) in 2005. From 2005 to 2007, he was an associate scientist at the Institute for Infocomm Research (I2R) in Singapore. Since 2007, he has been with UEF. In 2010-2012, his research was funded by the Academy of Finland in a post-doctoral project focusing on speaker recognition. He is the PI of a 4-year Academy of Finland project focusing on speaker recognition and a co-PI of another Academy of Finland project focusing on audio-visual spoofing. He chaired the latest Odyssey 2014: The Speaker and Language Recognition workshop, acts as an associate editor in IEEE/ACM Transactions on Audio, Speech and Language Processing and Digital Signal Processing. He also holds the honorary title of Docent at Aalto University, Finland, with specialization area in speaker and language recognition. He has authored about 100 peer-reviewed scientific publications in these topics.

Chin-Hui Lee is a professor at School of Electrical and Computer Engineering, Georgia Institute of Technology. Dr. Lee received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, in 1973, the M.S. degree in Engineering and Applied Science from Yale University, New Haven, in 1977, and the Ph.D. degree in Electrical Engineering with a minor in Statistics from University of Washington, Seattle, in 1981.

Dr. Lee started his professional career at Verbex Corporation, Bedford, MA, and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, where he engaged in research and product development in speech coding, speech synthesis, speech recognition and signal processing for the development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, New Jersey, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, adaptive and discriminative learning, biometric authentication, and information retrieval. From August 2001 to August 2002 he was a visiting professor at School of Computing, The National University of Singapore. In September 2002, he joined the Faculty of Engineering at Georgia Institute of Technology.

Prof. Lee has participated actively in professional societies. He is a member of the IEEE Signal Processing Society (SPS), and International Speech Communication Association (ISCA). In 1991-1995, he was an associate editor for the IEEE Transactions on Signal Processing and Transactions on Speech and Audio Processing. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. In 1995-1998 he was a member of the Speech Processing Technical Committee and later became the chairman from 1997 to 1998. In 1996, he helped promote the SPS Multimedia Signal Processing Technical Committee in which he is a founding member.

Dr. Lee is a Fellow of the IEEE, and has published close to 400 papers and 30 patents. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was awarded the prestigious Bell Labs President's Gold Award for his contributions to the Lucent Speech Processing Solutions product. Dr. Lee often gives seminal lectures to a wide international audience. In 2000, he was named one of the six Distinguished Lecturers by the IEEE Signal Processing Society. He was also named one of the two ISCA's inaugural Distinguished Lecturers in 2007-2008. He won the IEEE SPS's 2006 Technical Achievement Award for "Exceptional Contributions to the Field of Automatic Speech Recognition". He was one of the four plenary speakers at IEEE ICASSP, held in Kyoto, Japan in April 2012. More recently, he was awarded the 2012 ISCA Medal for "pioneering and seminal contributions to the principles and practices of automatic speech and speaker recognition, including fundamental innovations in adaptive learning, discriminative training and utterance verification."

Paper V

H. Behravan, T. Kinnunen, and V. Hautamäki
"Out-of-Set i-Vector Selection for Open-set Language Identification"
in Proc. of The Speaker and Language Recognition Workshop (Odyssey),
pp. 303–310, Bilbao, Spain, 2016.

Out-of-Set i-Vector Selection for Open-set Language Identification

Hamid Behravan, Tomi Kinnunen, Ville Hautamäki

School of Computing
University of Eastern Finland
{behravan,tkinnu,villeh}@cs.uef.fi

Abstract

Current language identification (LID) systems are based on an i-vector classifier followed by a multi-class recognition back-end. Identification accuracy degrades considerably when LID systems face open-set data. In this study, we propose an approach to the problem of out of set (OOS) data detection in the context of open-set language identification. In our approach, each unlabeled i-vector in the development set is given a per-class outlier score computed with the help of non-parametric Kolmogorov-Smirnov (KS) test. Detected OOS data from unlabeled development set is then used to train an additional model to represent OOS languages in the back-end. The proposed approach achieves a relative decrease of 16% in equal error rate (EER) over classical OOS detection methods, in discriminating in-set and OOS languages. Using support vector machine (SVM) as language back-end classifier, integrating the proposed method to the LID back-end yields 15% relative decrease in identification cost in comparison to using all the development set as OOS candidates.

1. Introduction

Language identification (LID) is the task of automatically identifying whether a known target language is being spoken in a given speech utterance [1]. Over the past years, several methods have been developed to perform LID tasks, including phonotactic [2] and acoustic ones [3]. The former uses phone recognizers to tokenize speech utterances into discrete units followed by n-gram statistics accumulation and language modeling back-end [4, 1]. The latter uses spectral characteristics of languages in a form of acoustic features such as shifted delta cepstral (SDC) coefficients [5, 6]. Gaussian mixture models (GMMs) [7] and support vector machines (SVMs) [8] are often used as classifiers. Recently, i-vectors [9] based on bottleneck features [10] have also been extensively explored.

State-of-the-art LID systems [11, 12, 10] achieve high identification accuracy in *closed-set* tasks, where the language of a test segment corresponds to one of the known target (in-set) languages. But in *open-set* LID tasks, where the language of a test segment might not be any of the in-set languages, accuracy often degrades considerably [13, 14]. In open-set LID, the objective is to classify a test segment into one of the pre-defined in-set languages or a single *out-of-set* (OOS) language (or model). Open-set LID is more applicable in real-life scenarios, where speech may come from any language. For example, in multilingual audio streaming and broadcasting, it is necessary to filter languages which do not belong to any of the modeled target languages [15].

Different approaches have been explored for OOS modeling both in open-set speaker identification (SID) and LID systems. In the context of open-set SID, the objective is to decide whether to accept or reject a speaker as being one of the enrolled speakers. The authors of [16, 17] used the knowledge of universal background model (UBM) to represent the OOS speakers. Each in-set speaker is modeled using Gaussian mixture models (GMMs) with a UBM and maximum a posteriori (MAP) speaker adaptation [18]. During classification, if any of the in-set speakers is selected, the test speaker is labeled as in-set; otherwise, the UBM has the highest score and the test speaker is classified as OOS. Authors in [19] proposed a system which first finds the best-matched model for a test speaker using vector quantization (VQ) based recognition system [20]. Then, a set-score is formed using support vector machine (SVM) classifier. Finally, a vector distance measurement for each enrolled speaker is used to accept or reject the test speaker as in-set class.

A few prior studies have been carried out on open-set LID tasks as well, as summarized in Table 1. To model OOS languages in open-set LID tasks, some approaches make use of additional OOS speech data derived from languages different from the target (in-set) languages [21, 22]. This additional OOS data is then used for training an OOS model. Obtaining additional data is often done in a supervised or semi-supervised way, which can be time consuming or leads to further sub-problems such as representative data selection to model the OOS languages. The authors of [21] proposed a method for compact OOS candidate language selection based on knowledge of world-language distance. In specific, candidate OOS data came from different language families having different prosody characteristics from the target languages. This method achieved 8.4% relative improvement in classification performance over a baseline system with a random selection of OOS candidates. In [22], a target-independent (TI) Gaussian was trained using development data of all target languages, to represent the OOS languages. Further, adopting maximum mutual information (MMI) [23] approach in [14] allows training an additional OOS Gaussian model using only in-set data. The trained OOS model improved the detection cost considerably, however, the impact of training the OOS model using actual OOS data was not investigated.

A practical key question in representing the OOS data is how to select the most representative OOS candidates to model OOS languages from a large set of unlabeled data. While random selection might be one option, in this study we attempt to specifically identify "higher quality" OOS utterances. To achieve this, we present a simple approach to find such OOS candidates in i-vector space [9], based on non-parametric Kolmogorov-Smirnov (KS) test [25, 26]. It gives each i-vector a per-class outlier score representing the confidence that an i-vector corresponds to an OOS language. This approach is fast and in contrast to [21, 22], requires no prior language labels of the additional data which may not be available in the real-world applications [27].

Table 1: Summary of the previous studies on open-set language identification task. Different approaches for selecting out-of-set (OOS) data include using in-set data [14], using all development data [22], selecting labeled out-of-set data [21], pooling additional data [13, 24] and finally selecting from unlabeled data (present study).

Study	Data	OOS selection	OOS modeling
Zhang and Hansen [21]	NIST LRE 2009	Supervised candidate selection	General Gaussian back-end
BenZeghiba *et al.* [22]	NIST LRE 2007	All development data as OOS	General Gaussian back-end
McCree [14]	NIST LRE 2011	No OOS detection	Gaussian discriminative training using in-set data
Torres-Carrasquillo *et al.* [13]	NIST LRE 2009	Additional OOS data	Several spectral and token classifiers
Torres-Carrasquillo *et al.* [24]	NIST LRE 2007	Additional OOS data	Several spectral and token classifiers
Present study	NIST i-vector challenge 2015	OOS selection from unlabeled data	Gaussian, cosine and SVM classifiers

2. Open-set language identification

In closed-set LID, the objective is to classify a test segment X into one of the pre-defined set of target (in-set) languages $\{L_m|m = 1, ..., M\}$, where M is total number of target languages. To classify X, the decision of the most similar language $\tilde{L}$ is chosen to maximize the *a posteriori* probability [28],

$$\tilde{L} = \underset{1\leq m\leq M}{\operatorname{argmax}}\, p(L_m|X) = \underset{1\leq m\leq M}{\operatorname{argmax}}\, p(X|L_m)p(L_m), \quad (1)$$

where the language likelihood $p(X|L_m)$ and language *a priori* probability $p(L_m)$ are assumed known. In *open-set* LID, the objective is to classify X into one of the $M+1$ languages, with M in-set languages and a single additional OOS language (or model).

Figure 1 shows a block diagram of a general open-set LID system used in this paper. OOS data selection block performs OOS detection on unlabeled development data to find best representative OOS data for training an additional OOS model.

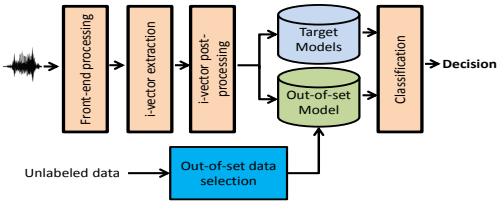


Figure 1: Block diagram of open-set language identification. The best-limited out-of-set (OOS) candidates are selected from the unlabeled development data for OOS modeling. We propose a simple method based on Kolmogorov-Smirnov test to find out-of-set data in the i-vector space.

In this study, we use i-vectors to represent utterances and consider the following three back-end language classifiers: **Gaussian** [22, 29], **cosine** [30] and **SVM** scoring [8, 31], to model both the target and the OOS languages. In the first case, for a given test i-vector, $\mathbf{w}_{\text{test}}$, the log-likelihood for a target language m is computed as

$$ll^m_{\mathbf{w}_{\text{test}}} = (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_m)^T\mathbf{w}_{\text{test}} - \frac{1}{2}\boldsymbol{\mu}_m^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_m \quad (2)$$

where $\boldsymbol{\mu}_m$ is the sample mean vector of target language m, and $\boldsymbol{\Sigma}$ is a shared covariance matrix common for all the languages. Having access to the training i-vectors, we form the maximum likelihood estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_m$'s, and use Eq. (2) to compute a language similarity score.

Cosine scoring is a dot product between test i-vector, $\mathbf{w}_{\text{test}}$, and language model mean, $\boldsymbol{\mu}_m$

$$\text{score}^m_{\mathbf{w}_{\text{test}}} = \frac{\mathbf{w}^T_{\text{test}}\ \boldsymbol{\mu}_m}{||\mathbf{w}_{\text{test}}||\ ||\boldsymbol{\mu}_m||}. \quad (3)$$

In addition, we used one-versus-all version of support vector machine (SVM) classifier with second order polynomial kernel [31] after experimenting with different kernel types. In the training phase, all samples of a target language and OOS languages are considered as positive instances with all the other languages corresponding to negative instances. The number of class separators equals the number of target languages plus one, the last coming from the OOS model. During testing phase, the highest score of a separator determines the class label of a test segment.

A simple LID baseline system is to treat the problem as a closed-set task without the OOS model. NIST has provided such a system in the download package of the recent 2015 language recognition i-vector challenge [32]. It is based on cosine scoring in which development data is used to estimate global mean and covariance to center and whiten the evaluation i-vectors. We will refer to this closed-set LID system as the **NIST baseline** in our results, in contrast to the open-set systems containing an additional OOS model.

3. Out-of-set data selection for OOS modeling

The objective in OOS detection is to assign each i-vector with an *outlier score*, higher value indicating higher confidence that the i-vector is an OOS observation (none of the known target languages). Since the main aim of this study is to select most representative OOS candidates to model OOS languages, we investigate three commonly used OOS detection methods, in general outlier detection context, as our baselines: (i) one-class SVM, (ii) k-nearest neighbour (kNN) and (iii) distance to cluster centroid. Each of these methods provides an outlier score for each of the scored unlabeled utterances. Then, for the purpose of OOS modeling (Figure 1), we apply 3-sigma-rule [33] for OOS selection, provided that the distribution of outlier scores for these three methods can be assumed normal.

3.1. One-class SVM

SVMs [34] are most commonly used as two-class classifiers. An SVM projects the data into a high-dimensional space and finds a linear separator between classes. In contrast, *one*-class SVM was proposed for out-of-set detection in [35]. In the training phase, the detector constructs a decision boundary to achieve maximum separation between the training points and the origin. A given unlabeled utterance is then projected into

the same high-dimensional space. The distance between the unlabeled utterance and the linear separator is used as the outlier score. We use LIBSVM[1] (version 3.21) to train an individual one-class SVM for each in-set class using the polynomial kernel [34] and the default parameters of the software package. The maximum score over in-set languages determines the outlier score for a given unlabeled utterance.

3.2. K-nearest neighbour (kNN)

In this technique, the outlier score for an observation is computed by the sum of its distances from its k nearest neighbours [36]. In this study, for each unlabeled utterance, the outlier scores are computed using k = 3 within each in-set language using Euclidean distance. Then, the maximum of outlier scores over all the in-set languages is used as the outlier score for that utterance.

3.3. Distance to class centroid

This is a simple classical approach to detect OOS data [37]. We assume that OOS data are far away from the class centroids. For instance, if the data follows a normal distribution, observations beyond two or three standard deviations above and below the class mean can be considered as OOS data [37]. This technique consists of two steps. First, the centroid of each in-set language is computed. Then, the distance between a data to the class centroid is computed as the outlier score. The maximum distance over all the in-set languages determines the outlier score for a given unlabeled utterance. We consider the in-set languages as different classes and the mean of each class as class centroids. Euclidean distance is chosen to compute the distance between each test data and the class means.

4. Proposed method

By now we have reviewed three commonly used OOS detection methods. Here we propose a simple and effective technique to find OOS data in the i-vector space. To this end, we adopt the non-parametric *Kolmogorov-Smirnov* (KS) test [25, 26]. It is used to decide whether a sample is drawn from a population with a known distribution (one-sample KS test) or to compare whether two samples have the same underlying distribution (two-sample KS test).

For any i-vector, $\mathbf{w}_i$, the distances of $\mathbf{w}_i$ to other i-vectors in language m has an empirical cumulative distribution function (ECDF) $F_{\mathbf{w}_i}(x)$ evaluated at x. The KS statistic between i-vector $\mathbf{w}_i$ and any other i-vector $\mathbf{w}_j$ in m can be computed by

$$\mathrm{KS}(\mathbf{w}_i, \mathbf{w}_j) = \max_x |F_{\mathbf{w}_i}(x) - F_{\mathbf{w}_j}(x)| \qquad (4)$$

Given language m with the total number of instances N, the outlier score for i-vector $\mathbf{w}_i$ is then defined as the average of these KS test statistics:

$$\mathrm{KSE}(\mathbf{w}_i) = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathrm{KS}(\mathbf{w}_i, \mathbf{w}_j) \qquad (5)$$

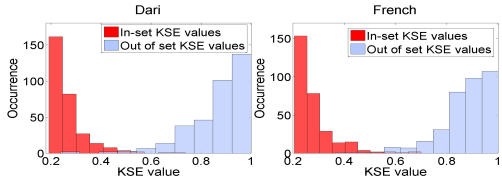
The average of KS statistics in Eq. (5), lies between 0 and 1; value close to 1 correspond to points with higher likelihood of being an OOS. Algorithm 1 shows a pseudo-code for computing the outlier score for a particular unlabeled i-vector.

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Algorithm 1 Outlier score computation for an unlabeled i-vector using KSE.

Let $L = \{l_1, l_2, \ldots, l_M\}$ be the set of M in-set languages
Let $W_m = \{\mathbf{w}_{m1}, \mathbf{w}_{m2}, \ldots, \mathbf{w}_{mN}\}$ be the set of i-vectors in in-set language l_m
Input $\mathbf{w}$ as an unlabeled i-vector
for $l_m \in L$ **do**
  $temp \leftarrow 0$
  for $\mathbf{w}_{mk} \in W_m$ **do**
    $KS \leftarrow$ compute KS value between $\mathbf{w}$ and $\mathbf{w}_{mk}$ using Eq. (4)
    $temp \leftarrow temp + KS$
  end for
  $KSE[m] \leftarrow$ divide $temp$ by $N - 1$, Eq. (5)
end for
$outlierscore \leftarrow$ Multiply $KSE[m]$ by -1 and select the maximum value

Figure 2 shows the distribution of per-language (in-set) and OOS KSE values for Dari and French. For inset values, only i-vectors of these languages were used to plot the distributions (in other words, i-vectors i and j in Eqs. (4) and (5) are both from the same languages). KSEs within each language have values close to zero. For the OOS KSE values in Figure 2, a set of i-vectors which do not belong to these languages were used to plot the same distributions (in other words, i-vector i does not belong to the language class of i-vector j, in Eqs. (4) and (5)). These i-vectors are considered as OOS to these languages. As expected, the KSE values tend to values close to 1.



(a) (b)

Figure 2: Distribution of in-set and OOS KSE values for two different languages, a) Dari and b) French. KSEs within each language have values close to zero. KSE values for OOS i-vectors tend towards one.

The Table 2 further demonstrates how we label data to evaluate our OOS detectors. Let us consider five i-vectors and their computed KSE values, given three in-set languages. The first three rows correspond to in-set utterances and the last two rows to OOS utterances. If the true language is one of the inset languages, label is set to 1 (e.g. the first row of Table 2), and to 0 otherwise (e.g. the last row of Table 2). The KSE values of each unlabeled utterance is multiplied by -1 and the maximum value is selected as the outlier score.

Following this method, we use box plot [33] to select OOS i-vectors. Box plot uses the median and the lower and upper quartiles defined as 25th and 75th percentiles. The lower quartile, median and upper quartile are often denoted by Q1, Q2 and Q3, respectively. In this study, unlabeled i-vectors with outlier scores above a threshold set at, $\mathrm{Q3} + 2.5 \times \mathrm{IQ}$, are selected for OOS modeling. Interquartile range or IQ denotes the difference $(\mathrm{Q3} - \mathrm{Q1})$.

Table 2: Example of test utterance labeling for the evaluation of OOS data detection task given multiple inset languages. KSE values for each data is computed according to Eq. (5).

Data_Id	True language	KSE values			In-set/OOS
		Greek	Dari	Urdu	
1	Greek	**0.29**	0.82	0.84	Inset
2	Dari	0.91	**0.11**	0.79	Inset
3	Urdu	0.85	0.92	**0.21**	Inset
4	Spanish	0.74	0.79	**0.64**	OOS
5	Farsi	0.81	**0.56**	0.77	OOS

5. Experimental set-up

5.1. Training, development and evaluation data

In this study, we used i-vectors provided by the National Institute of Standards and Technology (NIST) in their 2015 language i-vector machine learning challenge [32]. It is based on the i-vector system developed by the Johns Hopkins University Human Language Technology Center of Excellence in conjunction with MIT Lincoln Laboratory [6]. Table 3 shows the distribution of development, training and test sets provided in the challenge. The development set consists of 6500 unlabeled i-vectors intended for general system tuning. The training set consists of a set of 300 i-vectors for each of the 50 target languages, corresponding to 15000 training utterances in total. The test segments include 6500 unlabeled i-vectors corresponding to all of the target languages and an unspecified number of OOS languages. The i-vectors are of dimensionality 400.

Table 3: Distribution of training, development and test sets from the NIST 2015 language i-vector machine learning challenge. The i-vectors are derived from conversational telephone and narrowband broadcast speech data.

Dataset	#i-vectors	#languages	label
Training set	15000	50	labeled
Development set	6500	*n/a*	unlabeled
Test set	6500	50+OOS	labeled

To evaluate the OOS detection methods, we need both inset and OOS data. Since only the training set has labels, we further split it into three portions of non-overlapped utterances. We name them a development, training and test portion to make a distinction between them and the original training, development and test sets provided by NIST. Table 4 shows the distribution of these portions in our study. Training and development portions include non-overlapped utterances of same languages. These languages are called in-set languages and correspond to 30 different languages. Test portion consists of utterances corresponding to all of those 30 in-set languages plus utterances from 20 additional languages. We call these 20 languages as OOS languages and their corresponding utterances as OOS data.

Figure 3 further shows the Venn diagram illustrating the data overlap, i.e. individual utterances and languages, between training, development and test portions. The development portion is used for general OOS data detection tuning, such as parameter setting for one-class SVM and threshold setting to identify OOS data in the LID task. Training and test portions are used for building and evaluating OOS data detectors, respectively.

Table 4: Distribution of development, training and test portions for the out-of-set (OOS) data detection task. All portions are subsets of the original NIST 2015 LRE i-vector challenge training set.

	In-set	Out-of-set
Total number of languages	30	20
Count of dev. portion files	1500	—
Count of training portion files	6000	—
Count of test portion files	1500	6000

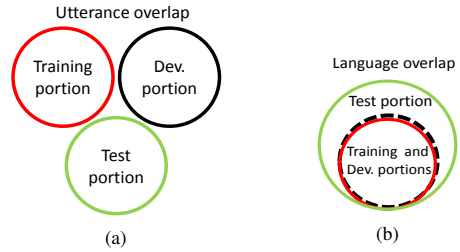


Figure 3: Venn diagram illustrating the data overlap between training, development and test portions, all being subsets of the original NIST 2015 LRE i-vector challenge training set. a) Utterance overlap. b) Language overlap.

5.2. i-Vector post-processing

The sample mean and the sample covariance of the unlabeled development set are computed to center and whiten all the i-vectors [38]. Then, length-normalization [38] is applied to project all the i-vectors onto a unit ball. These i-vectors are then further transformed using principal component analysis (PCA) [39], keeping the 99% of the cumulative variance. The resulting i-vector dimensionality is 391, just slightly smaller than original 400. Then, linear discriminant analysis (LDA) [6] is applied to reduce the dimensionality of i-vectors to the maximum number of classes minus one, in our case 49 dimensions. Following PCA and LDA, within-class covariance normalization (WCCN) [40] is used as a supervised transformation technique to further suppress unwanted within-language variation. For open-set LID, the projection matrices of PCA, LDA and WCCN are computed using the training set. The order of post-processing techniques follows the same order as [6].

5.3. Tasks and performance measure

We use detection error tradeoff (DET) curve [41] to evaluate the OOS data detection performance. It plots the false acceptance rate (FAR) versus false rejection rate (FRR), using a normal deviate scale. Here the task is to identify those test portion data which do not conform to any of the training portion classes.

The performance measure of open-set LID task as defined in the NIST 2015 language recognition i-vector challenge task is defined as follows [32]:

$$\text{Cost} = \frac{(1 - P_{\text{oos}})}{N} \sum_{k=1}^{N} P_{\text{error}}(k) + P_{\text{oos}} \times P_{\text{error}}(\text{oos}) \quad (6)$$

where $P_{\text{error}}(k) = (\frac{\#errors\_class\_k}{\#trials\_class\_k})$, $N = 50$, and $P_{\text{oos}} = 0.23$.

Open-set LID is performed using the training and test sets (not portions) for model training and evaluation, respectively. Detected OOS data from the development set is then used for OOS modeling.

6. Results

In order to evaluate our proposed OOS selection method, we separate the experiments into two parts. First, we evaluate the performance of the proposed and the baseline OOS detectors. Then, we assess the open-set language identification task with the OOS model trained by the additional data selected by our proposed OOS detector.

6.1. Stand-alone out-of-set data selection

Figure 4 shows the impact of parameter k on kNN-based OOS detection task, in terms of EER. As shown, no considerable change is observed by changing the value of k. For the remaining experiments, we arbitrarily fix $k = 3$.

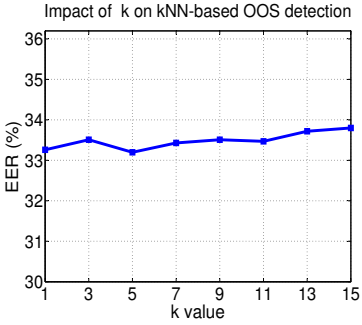


Figure 4: Impact of k value on kNN-based OOS detection method in terms of EER (%). No improvement is observed by increasing k.

Next, as KSE is based on the distribution of distances between points, here we study the impact of different distance metrics [42] on our proposed OOS detector. To this end, we vary the distance metric used in computing ECDFs in Eq. (4). Figure 5 shows the results. Euclidean and city-block distances achieve the highest performance with EERs of 28.80% and 28.46%, respectively. For the cosine distance with EER of 32.27% and Pearson correlation distance with EER of 32.35%, the performance degradation is more pronounced. For the remaining experiments, we fix the Euclidean distance metric.

Figure 6 shows the DET curve comparison between the proposed KSE and the three baseline methods on the test portion data. The results indicate that the proposed KSE method outperforms the baselines in terms of EER. KSE outperforms kNN and one-class SVM by 14% and 16% relative EER reductions, respectively. From the baselines, the distance to class mean method obtains the lowest performance with EER of 36.34%.

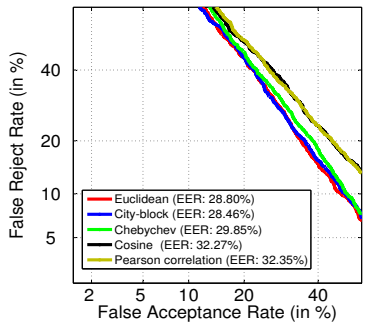


Figure 5: Performance of proposed OOS detection method under different distance metrics. Euclidean and city-block distances achieve the highest performance.

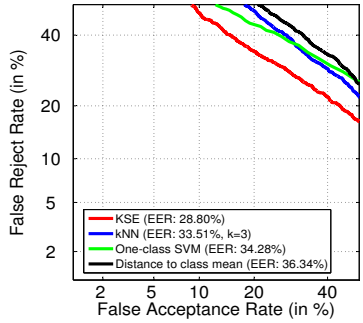


Figure 6: Comparison between the performance of the proposed OOS detection and three different baseline methods. KSE method shows the best performance compared to baseline methods.

Now we turn our attention to the effects of system fusion on the OOS detection performance at the score level. To this end, we adopt linear score fusion function optimized with the cross-entropy objective [43] using the BOSARIS Toolkit [44]. We use our development portion to find optimal classifier weights. Figure 7 shows the results of fusion of KSE to baseline OOS detection methods (2-way score fusion) on the test portion data. Interestingly, fusion of KSE to baseline systems improves the accuracy substantially. A relative decrease of 27% over KSE is achieved by fusing KSE and one-class SVM, yielding EER of 20.93%. EER of 28.25% is obtained by fusing KSE and kNN, yielding relative decrease of 2% and 16% over KSE and kNN, respectively. Fusion of all four methods (4-way score fusion) achieves EER of 22.09%, i.e. relative decrease of 23% and 27% over KSE and fusion of all baseline OOS detection methods (3-way score fusion), respectively.

6.2. Language identification

Up to this point, we have discussed OOS data detection accuracy. Now we turn our attention to the full language identification system in Figure 1 with the OOS model being trained by the additional data selected using one of the OOS detection methods. Table 5 shows the identification results for both closed-set

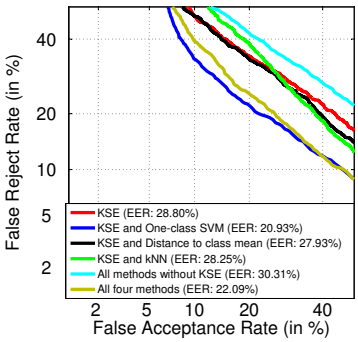


Figure 7: Fusion of KSE to baseline OOS detection methods. Fusion of KSE to one-class SVM yields the best performance. "All methods without KSE" refers to the score fusion of one-class SVM, kNN, and distance to class mean methods (3-way score fusion). "All four methods" indicates fusion of all four methods including KSE (4-way score fusion).

and open-set LID systems for different classifiers. The rows of Table 5 differ based on the data selected for target-independent OOS modeling. Rows 2 and 3 correspond to systems in which the data of *all* the training and development sets are used for OOS modeling, respectively, inspired by the work described in [22]. Row 4 corresponds to pooling the data in both development and training sets. The proposed selection method refers to OOS data selection using KSE. Finally, for reference purposes, the last row shows the closed-set results, where the language of a test segment can be only one of the target languages (no OOS modeling is performed).

We observe, firstly, that integrating the proposed selection method to open-set LID system with an SVM language classifier outperforms the other systems. The lowest identification cost achieved, 26.61, outperforms the NIST baseline system by 33% relative improvement. Using all the training or development data for OOS modeling does not necessarily lead to considerable improvement over closed-set results. For example, taking Gaussian scoring and training data (second row), identification cost decreases from 37.07 in closed-set to 34.15 in open-set, yielding a relative improvement of 8%. Furthermore, assuming an open-set LID system based on random OOS data selection (first row), the proposed method achieves a relative improvement of 17% using SVM language classifier[2]. It is worth mentioning that since test set contains a considerable amount of OOS utterances, the closed-set LID system incorrectly classifies them as one of the target languages. This explains why the closed-set LID system generally underperforms the open-set one.

Now, we treat the open-set LID as a binary classification task, discriminating in-set and OOS test data. In-set test data refers to those test files having the same language labels as one of the target languages. Table 6 shows the confusion matrix of in-set/OOS classification using KSE and three different language classifiers. Results correspond to the fifth row of Table 5. The results indicate that from 1500 OOS test data, 1012 are classified correctly as OOS using KSE with SVM language classifier. This number is 317 and 49 for Gaussian and cosine scoring, respectively.

[2]Training the OOS model with known identities of OOS languages was not possible since NIST had not provided the class labels of the development set utterances.

Table 5: Language identification results for both open-set and closed-set set-ups. Rows differ based on the data used for OOS modeling. Results are reported based on identification cost, lower cost indicating higher performance. Numbers in parentheses indicate amounts of selected data for OOS modeling. The results are reported from the evaluation online system provided by the NIST in the i-vector challenge.

Data selected for OOS modeling (#)	Cosine	Gaussian	SVM
Random (1067)	36.25	34.20	32.11
Training (15000)	36.35	34.15	32.61
Development [22] (6431)	36.10	32.87	31.23
Training+Dev. (21431)	36.46	33.38	31.74
proposed selection method (1067)	34.28	32.23	**26.61**
Closed-set (no OOS model)	39.59*	37.07	37.23

*From the NIST baseline result

Table 6: Confusion tables for in-set and OOS classification using KSE with three different language classifiers corresponding to the fifth row of Table 5. In total, test set corresponds to 5000 and 1500 in-set and OOS data, respectively.

(a) SVM scoring

Pred. \ True	Inset	OOS
Inset	4134	866
OOS	488	1012

(b) Gaussian scoring

Pred. \ True	Inset	OOS
Inset	4867	133
OOS	1183	317

(c) Cosine scoring

Pred. \ True	Inset	OOS
Inset	4999	1
OOS	1451	49

Fixing SVM as the best language classifier, Table 7 compares the results of using different OOS detectors for OOS modeling in our open-set LID task. For comparison, we also include the closed-set LID results based on SVM in the last row of Table 7. The open-set LID system based on KSE outperforms the other methods, in terms of identification cost. Using KSE as an OOS detector brings relative improvements of 9% and 13% over kNN and one-class SVM, respectively.

Table 7: Open-set language identification results for different OOS data selection methods using SVM language classifier. Results are reported based on identification cost, lower cost indicating higher performance. The results are reported from the evaluation online system provided by the NIST in the i-vector challenge.

Method used for OOS modeling	Cost
KSE	**26.61**
kNN	29.33
One-class SVM	30.66
Distance to class mean	31.48
Closed-set	37.23

7. Conclusion

We focused on the problem of OOS data selection in the i-vector space in the context of open-set LID problem. We proposed an approach based on non-parametric Kolmogorov-Smirnov test to effectively select OOS candidates from an unlabeled development set. Our proposed OOS detection method outperforms the one-class SVM baseline by 16% relative improvement, in terms of EER. We then used OOS candidates to train an additional model to represent the OOS languages in the open-set LID task. The baseline system was realized by using all the development and/or training data as OOS candidates. Using SVM as language classifier, with the proposed OOS data selection method, identification cost was relatively decreased by 15% over using all the development set as OOS candidates in the open-set LID task.

In our future work, we plan to explore possible extensions of the Kolmogorov-Smirnov test, such as weighted Kolmogorov-Smirnov test, to improve OOS detection accuracy. In addition, we will investigate clustering and modeling of KSE scores.

8. References

[1] K.J. Han and J. Pelecanos, "Frame-based phonotactic language identification," in *Proc . of SLT*, 2012, pp. 303–306.

[2] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proc. of ACL*, 2005, pp. 515–522.

[3] N. Brümmer, A. Strasheim, V. Hubeika, P. Matějka, L. Burget, and O. Glembek, "Discriminative acoustic language recognition via channel-compensated GMM statistics," in *Proc. of INTERSPEECH*, 2009, pp. 2187–2190.

[4] J. L Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices," in *Proc. of INTERSPEECH*, 2004, pp. 1283–1286.

[5] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 271–284, 2007.

[6] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. of INTERSPEECH*, 2011, pp. 857–860.

[7] 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 *Proc. of SLP*, 2002, pp. 89–92.

[8] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A., "Language recognition with support vector machines," in *Proc. of Speaker Odyssey*, 2004, pp. 41–44.

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

[10] Y. Song, Xinhai Hong, B. Jiang, R. Cui, I. Vince McLoughlin, and L.-R. Dai, "Deep bottleneck network based i-vector representation for language identification," in *Proc. of INTERSPEECH*, 2015, pp. 398–402.

[11] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. of ICASSP*, 2014, pp. 5337–5341.

[12] M. Van Segbroeck, R. Travadi, and S.S. Narayanan, "Rapid language identification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 23, no. 7, pp. 1118–1129, 2015.

[13] P. A. Torres-Carrasquillo, E. Singer, T. P. Gleason, A. McCree, D. A. Reynolds, F. Richardson, and D. E. Sturim, "The MITLL NIST LRE 2009 language recognition system," in *Proc. of ICASSP*, 2010, pp. 4994–4997.

[14] A. McCree, "Multiclass discriminative training of i-vector language recognition," in *Proc. of Speaker Odyssey*, 2014, pp. 166–171.

[15] M. Adda-Decker, *Automatic Language Identification*, In Spoken Language Processing, J.J. Mariani ed., Wiley-ISTE, Chapter 8, 2009.

[16] J.H. L. Hansen, J.-W. Suh, and M. R. Leonard, "In-set/out-of-set speaker recognition in sustained acoustic scenarios using sparse data," *Speech Communication*, vol. 55, no. 6, pp. 769–781, 2013.

[17] P. Angkititrakul and J.H.L. Hansen, "Discriminative in-set/out-of-set speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 498–508, 2007.

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[19] J. Deng and Q. Hu, "Open set text-independent speaker recognition based on set-score pattern classification," in *Proc. of ICASSP*, 2003, vol. 2, pp. II–73–6 vol.2.

[20] J. Pelecanos, S. Myers, S. Sridharan, and V. Chandran, "Vector quantization based Gaussian modeling for speaker verification," in *Proc. of ICPR*, 2000, vol. 3, pp. 294–297 vol.3.

[21] Q. Zhang and J. H. L. Hansen, "Training candidate selection for effective rejection in open-set language identification," in *Proc. of SLT*, 2014, pp. 384–389.

[22] M.F. BenZeghiba, J. Gauvain, and L. Lamel, "Gaussian backend design for open-set language detection," in *Proc. of ICASSP*, 2009, pp. 4349–4352.

[23] L. Burget, P. Matejka, and J. Cernocky, "Discriminative training techniques for acoustic language identification," in *Proc. of ICASSP*, 2006, vol. 1, pp. I–I.

[24] P. A. Torres-Carrasquillo, E. Singer, T. P. Gleason, A. McCree, D. A. Reynolds, F. Richardson, W. Shen, and D. E. Sturim, "The MITLL NIST LRE 2007 language recognition system," in *Proc. of INTERSPEECH*, 2008, pp. 719–722.

[25] N. V. Smirnov, "Estimate of deviation between empirical distribution functions in two independent samples," *Bulletin Moscow University*, pp. 2: 3–16, 1933.

[26] M.S. Kim, "Robust, scalable anomaly detection for large collections of images," in *Proc. of SocialCom*, 2013, pp. 1054–1058.

[27] H. Lee, *Unsupervised Feature Learning Via Sparse Hierarchical Representations*, Stanford University, 2010.

[28] M.A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 4, no. 1, pp. 31–34, 1996.

[29] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Language score calibration using adapted Gaussian backend.," in *Proc. of INTERSPEECH*, 2009, pp. 2191–2194.

[30] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. of INTERSPEECH*, 2009, pp. 1559–1562.

[31] S. Yaman, J. W. Pelecanos, and M. K. Omar, "On the use of non-linear polynomial kernel svms in language recognition," in *Proc. of INTERSPEECH*, 2012, pp. 2053–2056.

[32] "The 2015 language recognition i-vector machine learning challenge," `https://ivectorchallenge.nist.gov/`.

[33] M. Natrella, *NIST/SEMATECH e-Handbook of Statistical Methods*, NIST/SEMATECH, chapter 7, 2010.

[34] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, 2008.

[35] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1472, 2001.

[36] J. Zhang and H. H. Wang, "Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance," *Knowledge and Information Systems*, vol. 10, no. 3, pp. 333–355, 2006.

[37] J.R. Kornycky and D.C. Ferrier, *Method for determining whether a measured signal matches a model signal*, Google Patents, 2014.

[38] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. of INTERSPEECH*, 2011, pp. 249–252.

[39] W. Rao and M.-W. Mak, "Alleviating the small sample-size problem in i-vector based speaker verification," in *Proc. of ISCSLP*, 2012, pp. 335–339.

[40] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of INTERSPEECH*, 2006.

[41] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," in *Proc. of EUROSPEECH*, 1997, pp. 1895–1898.

[42] E. Deza and M.-M. Deza, *Dictionary of Distances*, Elsevier, 2006.

[43] N. Brümmer, L. Burget, J. H. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[44] N. Brümmer and E. de Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing,," in *Technichal Report*, 2011, p. [Online]. Available: https://sites.google.com/site/nikobrummer.