COMPARISON OF TEXT-INDEPENDENT SPEAKER VERIFICATION SYSTEMS IN A MULTI-CLASS, SEMI-AUTOMATIC DETECTION SCENARIO

A Thesis

by

Fatih Yeşil

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
June 2013

COMPARISON OF TEXT-INDEPENDENT SPEAKER VERIFICATION SYSTEMS IN A MULTI-CLASS, SEMI-AUTOMATIC DETECTION SCENARIO

Approved by:

Assistant Professor Cenk Demiroğlu,
Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Assistant Professor Barış Aktemur
Department of Computer Science
Özyeğin University

Assistant Professor Fatih Uğurdağ
Department of Electrical and Electronics Engineering
Özyeğin University

Date Approved: 12 June 2013

To my family

ABSTRACT

Performance of the speaker verification systems is typically measured based on their binary decision accuracy. Soft outputs of the systems are used mostly for calibration or multiple system combination purposes. However, in speaker verification applications where close to 100% accuracy is required, such as the systems that are used in the call centers of finance companies, it is not possible to rely on the binary decisions of the existing verification systems. Still, in such cases, multi-class verification outputs (for example, high, medium and low verification score) returned by the speaker verification systems can be used by a human agent to either reduce the verification time and/or increase the verification accuracy compared to a human-only scenario.

In this thesis, an overview of a speaker verification system is given explaining in detail the algorithms that are implemented. Particularly the details about a classifier, GDA, which was firstly used by us for a verification purpose are given. It does relatively better job than state of the art algorithms for non-linear data like in our case. In the experiments section, some of the most popular speaker verification systems are compared in terms of the classical performance metric used in the literature. Then, multi-class output performance of them is compared when a human agent is assumed to be in the verification loop. Performance is measured by the reduction in the number of questions used by the human agent for verifying the identity of the caller without compromising the security. Experiments are performed using the NIST 2006 and 2008 databases. Eight and one conversation sides (5 minutes each) enrollment data and 1 side and 10 seconds verification data conditions are used.

ÖZETÇE

Konuşmacı doğrulama sistemlerinin başarısı tipik olarak bu sistemlerin ikili karar vermedeki doğruluğuna dayanarak ölçülür. Sistemlerin gerçek değerli çıktıları çoğunlukla sistem kalibrasyonu veya çoklu sistem kombinasyonları gibi amaçlar doğrultusunda kullanılır. Ancak, finans firmalarının çağrı merkezleri gibi yerlerde kullanılan ve %100'e yakın kesinlik gerektiren konuşmacı doğrulama uygulamalarında, varolan sistemlerin ikili kararlarına güvenmek mümkün değildir. Yine de bu tür durumlarda, konuşmacı doğrulama sistemi tarafından döndürülen çoklu-sınıf doğrulama çıktıları (örneğin yüksek, orta, düşük doğrulama yüzdesi) çağrı merkezi temsilcisi tarafından, sadece insan olan senaryoya göre doğrulama süresini kısaltmak ve/veya doğrulama kesinliğini arttırmak için kullanılabilir.

Bu tezde ilk olarak gerçekleyip kullandığımız algoritmaları detaylı bir şekilde anlatarak bir konuşmacı doğrulama sisteminin genel görünümünü vereceğiz. Bilhassa bir doğrulama amacı için ilk defa bizim kullandığımız, bir sınıflandırıcı olan GDA hakkında detaylı bilgi vereceğiz. GDA bizim ele aldığımız problemdeki gibi doğrusal olmayan verilerin sınıflandırılmasında görece olarak daha iyi çalışıyor. Deneyler bölümünde ise öncelikle bazı çok bilinen konuşmacı doğrulama sistemlerinin başarımlarını klasik başarım ölçütlerini kullanarak karşılaştırdık. Daha sonra, doğrulama döngüsünde bir çağrı merkezi temsilcisinin de olduğunu varsayarak, bu sistemlerin çoklu-sınıf başarımlarını karşılaştırdık. Başarım, temsilcinin güvenlikten ödün vermeden sorması gereken soru miktarındaki azalmaya göre ölçüldü. Deneyler NIST 2006 ve 2008 veritabanları kullanılarak gerçekleştirildi. Herbiri beşer dakikalık olan bir ve sekiz karşılıklı konuşmadan alınan kayıtlar ses imzalarının çıkarımında kullanıldı.

Doğrulama yapılacak konuşma içinse beş dakikalık bir ve on saniyelik bir kayıt kullanıldı.

ACKNOWLEDGEMENTS

I would like to thank my advisor Cenk Demiroğlu for his guidance and support throughout my master study. Also, I would like to thank my lab mates, particularly Ekrem Güner and Abdullah Erdoğan, for their friendship and support in any problem that I encountered.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Recognition lexically is the agreement that something is true or legal[1]. In speech processing domain, speaker recognition is the detection of a person by means of his/her voice. Recognition of a speaker using voice signature can be classified in two groups of research topic: speaker identification and speaker verification. In speaker identification problem, there is a pool of target speakers and the system tries to determine the identity of the trial speaker by matching the most possible one from the speaker set. The system may reject the trial if it gets a relatively low score below the threshold trained previously. On the other hand, speaker verification is the decision process of whether a trial speaker it matches the claimed id or not. To verify a speaker, that person should have a voice signature trained and kept in the database.

1.1 Overview of a Speaker Verification System

Basically a verification scenario consists of two main parts: training and decision. In Figure 1, the steps of training phase which is offline, is illustrated. Likewise the training decision phase is online and it is illustrated in Figure 2.

Speaker verification is basically a binary classification problem where each speaker is assumed to be a single class. To make a decision between two classes, firstly discriminative features should be defined using raw speech. In speaker verification frequently used features are *short-term spectral features*. They are called low-level features too,

[1]http://dictionary.cambridge.org

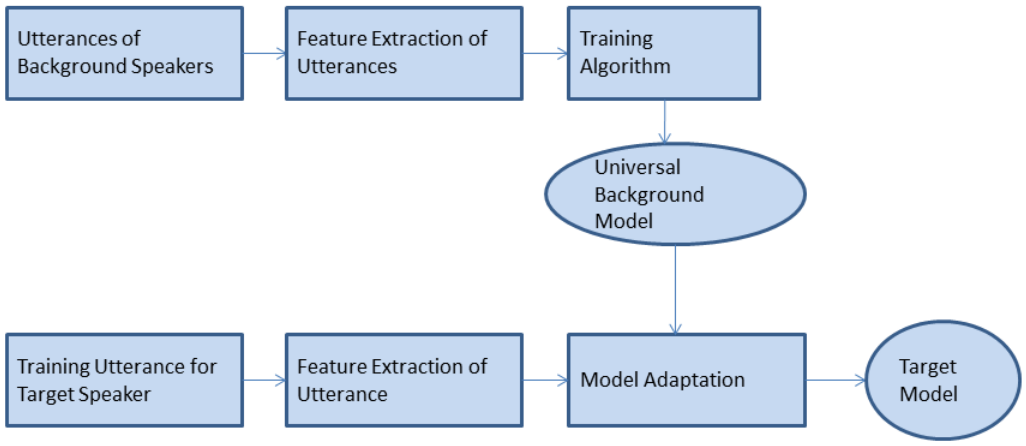

Figure 1: Training phase of a verification system. It is a time consuming process performed offline. Gathering the proper database, training the parameters, the threshold tuning and calibration of the system are all done during this phase.

which will be given in Section 8.1.1 in detail. Some well known examples are *mel-frequency cepstral coefficients* (MFCC), *linear predictive cepstral coefficients* (LPCC), *perceptual linear prediction* (PLP) coefficients, *line spectral frequencies* (LSFs) [1]. Proper features are extracted and generally delta and delta-delta features are also used within the static features.

Silence parts are removed using a *voice activity detector* (VAD). Energy based classification is one of the detector type. In short-term analysis, energy of each frame is computed then a relative thresholding may be applied. For example the frames that are 30dB below the frame with maximum energy are discarded. Histogram method is another way of silence removing. Two Gaussians are fit on to the energy

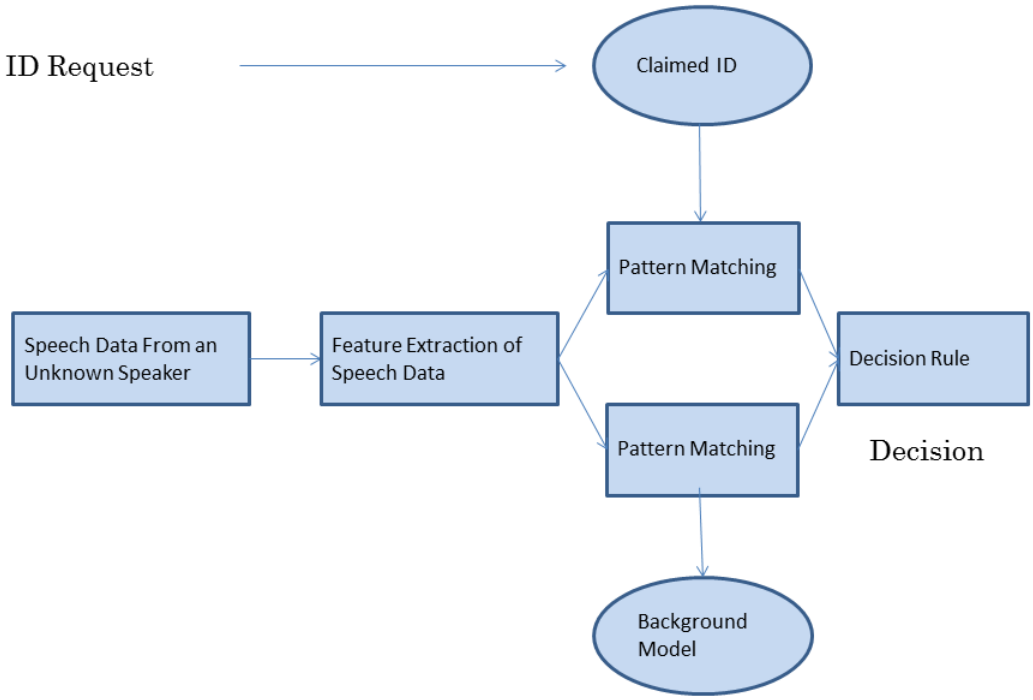

Figure 2: Decision phase of a verification system. It is done online. Likelihood of data captured from the current speaker is computed using the previously trained claimed ID model. A zero-one decision after thresholding or a probability is returned.

values and the component with higher energy is labeled as speech. Further processing may also be needed to avoid from keeping the high-energy noise frames. *Long-term spectral divergence* (LTSD) method and *periodicity based* VAD are other options to be used [1].

Time domain signal can be enhanced to suppress the noise effects. Due to computational loads of time domain analysis feature domain normalizations are done [1]. *Cepstral mean normalization* (CMN),also called *cepstral mean subtraction* (CMS) and *cepstral variance normalization* (CVN) are simple and well known normalization methods. Mean of each feature is subtracted from each frame and then each frame is divided by the standard deviation of that feature. This can be applied globally or locally using

a sliding window (mostly 3 second windows are used in literature). *Feature warping*, *short-time Gaussianization* and *feature mapping* are other channel compensation techniques that can be applied in feature domain. *Feature warping* and *short-time Gaussianization* both try to convert the cumulative distribution of frames to a target distribution. 3s length of sliding windows are used generally and for each feature vector warping is applied independently before delta features are computed. This independency brings a high computational load. In *feature mapping* labels of channel type,i.e. landline, cell, cordless, are needed. For each type of channel a unique GMM is adapted by MAP adaptation from a more general GMM. According to the likelihood proper MAP adapted GMM is used for an unknown channeled utterance.

After getting ready the features, next step is speaker modeling. There are two types of methods dealing with the model generation: discriminative and generative methods. Discriminative methods, like *support vector machine*(SVM) and *artificial neural networks* (ANN) [2], try to separate two classes as much as possible. In SVM, the purpose is to obtain the proper hyperplane with the maximum margin. Gaussian mixture model (GMM) is the mostly used, well known method which is a generative model. Unlike discriminative methods, generative methods are based on model fitting and computation of likelihoods instead of decision boundaries. For a text-dependent verification task *hidden Markov models* (HMMs) can be used for the likelihood functions since there exist a prior knowledge of what text will be spoken [2].

In general, session variability is the main problem of a speaker verification system. Channel mismatches, telephone type, handset type, background noise, record environment, mood of the speaker are the leading reasons causing the session variability. To remove these affects and build a robust system, there are some methods proposed both for generative and discriminative methods.*Joint factor analysis* (JFA) [3] is a successful method where the speaker mean supervectors (GMMs for each speaker) are assumed to be linear combinations of channel and speaker variabilities. They are

called channel and speaker factors. Estimation of these factors are done jointly. As a discriminative method, for SVM, *nuisance attribute projection* (NAP) and *within-class covariance normalization* (WCCN) are proposed methods to compensate the session variabilities. They will be both in detail given in Chapter 5.

Calibration of the system is an another significant issue which is related with scoring and thresholding namely back-end section. For example NIST provides development data for preparation of the system and training of the parameters to be used in the application. The developed system parameters and thresholds may not work as good as expected. Each speaker may have a different threshold that separates the target speaker from its impostors. Some sort of score normalizations are proposed to overcome these problems. *ZNorm*, *TNorm*, *ZTNorm* [2] are frequently used ones. In *ZNorm* obtained likelihood score ls is normalized as $\widehat{ls} = \frac{ls-\mu_I}{\sigma_I}$. The parameters μ_I and σ_I are computed offline for each speaker using a cohort of impostors.

1.1.1 i-vector based systems

I-vector based systems have recently become the dominant approach in speaker recognition area and they are constructed on Gaussian mixture models. I-vector extraction was introduced by Dehak [4]. Since the variabilities among the individuals and sessions of individuals are taken into account together instead of modeling them separately as in JFA, this approach is also called *total variability space*(TVS) approach. Extracted i-vectors that are used as the identities of individuals have lower dimensions than the conventional GMM supervectors. Besides it provides channel compensation. Dimensionality reduction reduces the computational load so that further improvements follow the i-vector method and it leads the *National Institute of Standards and Technology*(NIST) to raise the number of trials and present new challenges in the speaker recognition evaluations. Details about the speaker recognition evaluations conducted by NIST will be given in Chapter 2.

Assuming that the i-vectors have Gaussian distribution *probabilistic linear discriminant analysis*(PLDA) is used [5], which is firstly introduced for face recognition task. System performances gained an improvement via PLDA which will be given in detail in Section 5.2. Since the real i-vectors are not Gaussian due to the outliers, Kenny [6] proposed *heavy tailed probabilistic linear discriminant analysis*(HT-PLDA) where the prior distributions are chosen as *Student's t* instead of *Gaussian*. Using the heavy-tailed priors, the effect of outliers are aimed to reduce. Better performance is obtained using this new prior assumptions. Although HT-PLDA shows better performance, its being computationally expensive still remains a problem. Starting from this point, in [7] it was shown that using a nonlinear transformation non-Gaussian behaviors of i-vectors can be eliminated. Instead of defining a new distribution for the original i-vectors converting them to Gaussian distributed i-vectors provides the similar performance as HT-PLDA gets with less complexity in computations.

1.2 About this thesis

Speaker verification is becoming a widely deployed technology in many real-life applications such as call centers of financial institutions or telecom operators. Although very high accuracies are obtained in text-dependent tasks, text-independent verification is still problematic especially when the amount of verification data is very small. Luckily, in many application scenarios, such as the call centers of financial institutions, multiple sessions are available for enrolling a speaker which helps improve the performance. At the OZU speech lab, we are focused on such text-independent verification problems when multiple enrollment sessions are available. To that end, several popular speaker verification algorithms have been implemented and compared. Moreover, as a novel approach, the use of generalized discriminant analysis (GDA) [8] for speaker verification have been investigated and it has been found that GDA outperforms the alternative techniques in all cases.

Our work is based on the total variability space (TVS) approach [4] at the core since TVS has been found to be the dominant technique in most of the recent literature. One of the goals of this work is to compare different classifiers for the identity vectors (i-vectors) that are produced by TVS. We have investigated cosine distance scoring [4], support vector machines (SVM) [9], and generalized discriminant analysis (GDA) as classifiers. It is well-known that inter-session variability is an important problem that has substantial impact on the verification performance. As a second goal, we compared the performance of within-class covariance normalization (WCCN) [10], linear discriminant analysis (LDA) [4], nuisance attribute projection (NAP) [11], probabilistic LDA (PLDA) [5], and GDA algorithms in compensating for the intersession variability. Not only we investigated the classifiers and channel compensation schemes but also we compared their performances when coupled together.

Measuring the performance of the systems, we have found that none of the existing techniques have probability of false alarm, P_{FA}, that is low enough for some of the real-life applications such as the call centers of banks where speaker verification is perhaps most needed. Therefore, here, we investigate a semi-automatic verification approach where human agent is still in the loop but the load on the agent is reduced with the help of a multi-class automatic speaker verification system. Indeed, we have found that the automatic system can substantially reduce the number of security questions that an agent has to ask for verification without a sacrifice from security especially if the caller is a true client. Considering that the substantial number of calls in a typical call center application is made by true clients, the results indicate huge potential reduction in the amount of time needed for verification over the phone.

1.2.1 Outline of this thesis

This thesis is organized as follows. In Chapter 1 definition of a speaker verification system and an overview of it are given. In Chapter 2 NIST is introduced and details about the evaluations conducted by NIST are given. In Chapter 3 universal background modeling is introduced and details about the training phase is given.

In Chapter 4 total variability space approach is described. Since it has been the dominant approach in this research area recently, i-vector extraction steps are given in detail. In Chapter 7, we give the generalized discriminant analysis approach that we proposed the usage of it in the speaker verification domain. Channel compensation algorithms that are used here are described in Chapter 5.

In Chapter 6 different scoring techniques and performance metrics are given. We also propose semi-automatic approach and describe performance metric used within this new approach in this chapter. Experimental results are presented in Chapter 8. All details about the set-up, dimensions and datasets are described in Section 8. Conclusion is done in Chapter 9

CHAPTER II

NIST SPEAKER RECOGNITION EVALUATIONS

Speaker Recognition Evaluations have been held by a speech group at National Institute of Standards and Technology (NIST) since 1996. The purpose of these SRE's is to contribute to the direction of research and the calibration of the existing text-independent speaker recognition systems. It is open to all interested sides, researchers from universities and industries.

Evaluation period starts with the announcement of upcoming evaluation plan that gives of information about the tasks, rules and deadlines. After registration, firstly development data is sent to participants to make them train and calibrate their systems. Then evaluation data is sent and they are asked to submit the results in a limited time period. The sides are also asked to submit the description of their system including the algorithms used, execution time per CPU, computational sources, etc. Eventually the evaluation ends up with a workshop where the official results are declared and sides present their findings. By means of the new findings, algorithms or problems encountered next evaluation's main point starts to be figured out. The current plan of NIST is to organize this evaluation biyearly which was conducted annually till 2006.

The basic test in NIST SRE's has been speaker detection since 2004, i.e., to decide whether a known target speaker is speaking during a given segment of speech or not. Previously speaker segmentation, speaker tracking were also asked to be done.

The storage format for all the utterances is $8-bit\ \mu-law$ encoded speech signals that are sampled at $8-kHz$ sampling rate. The utterances are kept in SPHERE[1]

[1] $ftp://jaguar.ncsl.nist.gov/pub/sphere_2.6a.tar.Z$

files separately. The header file of a SPHERE file contains record information(i.e., phone call, phone call recorded over microphone, interview recorded over microphone), sample count, channel count, sampling rate, encoding type etc. The language spoken in the utterance is also provided in the header. Until 2012 SRE *automatic speech recognition* (ASR) transcriptions have also been provided to the participants for the records in English. All these informations are allowed to be used.

2.1 General Evaluation Rules

- Only the informations of specified trial and claimed speaker can be used. Any other test segments information can not be used except for the unsupervised adaptation mode which is given in Section 2.2.1. For instance: the use of evaluation data for impostor modeling in a normalization purpose.

- If an unsupervised adaptation mode is used, the process order of the test segments should be considered.

- Manually created transcriptions or any other information can not be used.

- Gender information is provided and it is allowed to be used. Anyway there is no cross gender test segment.

- Information about the telephone transmission channel type(i.e., landline, cordless, cell phone) and of the telephone instrument type(i.e., speaker-phone, earbud, head mounted) used in all segments is not provided and allowed. If they are determined by automatic means, they can be used.

- Listening the records or any other interaction, testing is not allowed.

- Any information given in the header part of each utterance is allowed.

2.2 Milestones in NIST SRE's through Training and Test Conditions

Since 2004 SRE is an important milestone and currently used NIST database by sides start from 2004, a brief information will be given by 2004.

2.2.1 2004 SRE

In 2004, new conversational speech data was used. This data was collected in the mixer project where Linguistic Data Consortium's new "'Fishboard"' platform was used. This database mostly consisted of conversational telephone speech in English, but it had some speech in non-English languages and some data recorded over microphone. Previous evaluations primarily concentrated on either regular telephone data or cellular phone data. In 2004 SRE both of them were utilized for the tasks.

For SRE'04 speaker detection is the basic test as in the last eight years from 1996. The evaluation included twenty-eight different speaker detection tests named by the amount and type of data both in train and test segments. These twenty-eight tests composed of the combinations of seven train and four test segments. One of them is the obligatory test to be submitted by the participants. Table 1 shows the possible combinations, the required one is typed in bold font.

- Training Conditions

  Unlike previous years there was no prior removal of intervals of silence. While for the past evaluations regular phone or cellular phone data had the priority, in 2004 evaluation both of them was the interest point.

  1. A sample from a single channel conversation side which is approximately 10 seconds speech
  2. A sample from a single channel conversation side which is approximately 30 seconds speech

3. Single channel conversation side which has approximately five minutes total duration including silence.[2]
4. Three single conversation sessions that belong to the same speaker
5. Eight single conversation sessions that belong to the same speaker
6. Sixteen single conversation sessions that belong to the same speaker
7. Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker who is the target speaker for all three summed and a second person who is different for each of them

- Test Conditions

1. A sample from a single channel conversation side which is approximately 10 seconds speech
2. A sample from a single channel conversation side which is approximately 30 seconds speech
3. Single channel conversation side which has approximately five minutes total duration including silence.
4. A summed-channel conversation, created concatenating the two sides of existing conversations sample by sample.

participants were asked to complete core condition, 1side-1side, which means the target speaker has one utterance for enrollment and one for testing, both are ∼5 min. durations. Remaining 27 conditions were up to the sides whether to submit or not. All training and test segment condition pairs are given in Table 1. What to submit

[2]Each conversation side has the last five minute excerpt of an approximately six-minute conversation. By means of this, beginning part of the conversation which seems to be less informative is eliminated. Silence removal is not done anymore, it contains silence

for each trial of given train-test pairs was to decide whether 'true' or 'false' within the likelihood score as a system confidence. The participants were also allowed to submit their unsupervised adaptation mode results where the target models can be updated using the previously evaluated trials that are determined as clients. Unsupervised adaptation mode was allowed for the first time in 2004 SRE.

		Test Segment Condition			
		10 sec	30 sec	1 side	1 conv
Training Condition	10 sec	optional	optional	optional	optional
	30 sec	optional	optional	optional	optional
	1 side	optional	optional	**required**	optional
	3 sides	optional	optional	optional	optional
	8 sides	optional	optional	optional	optional
	16 sides	optional	optional	optional	optional
	3 convs	optional	optional	optional	optional

Table 1: Enrollment and test conditions for SRE'04. The bold typed area is the required condition of the evaluation. The rest is up to the participants.

2.2.2 2005 SRE

One year later, in 2005, there was a big change. For the first time both side of all two channel conversations were provided to the sides. The purpose was to aid systems in echo cancellation and dialog analysis. Data was collected in the mixer project where Linguistic Data Consortium's new "'Fishboard"' platform, same as the previous year, and additionally some "'multi-channel"' data recorded at the same time from different microphones. The records were mostly in English. 20 different conditions were provided to be tested but one of them was obligatory, 1conv-1conv as shown in

Table 2. For 2005 SRE summed-channel tests were identical to those of 2004 SRE, so the results can be compared to see the improvement of the system performance from the previous year if desired. But for the rest there are some significant changes made.

- Training Conditions

  Unlike pre-2004 years there was no prior removal of intervals of silence from the utterances. For the two channel sided segments, the identity of the target speaker is given.

  1. A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)
  2. One two-channel conversation segment which has approximately five minutes total duration including silence
  3. Three two-channel conversation segment, each has approximately five minutes total duration including silence
  4. Eight two-channel conversation segment, each has approximately five minutes total duration including silence
  5. Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

- Test Conditions

  As it will be mentioned soon, auxiliary microphone data was also new for 2005.

1. A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)

2. One two-channel conversation segment which has approximately five minutes total duration including silence

3. A summed-channel conversations, created concatenating the two sides of existing conversations sample by sample.

4. One two-channel conversation segment where the usual telephone speech segment was replaced by microphone data for the target speaker side. This microphone data was provided with 8 kHz sampling rate and 8-bit $\mu - law$ encoding form.

		Test Segment Condition			
		10 sec 2-chan	1 conv 2-chan	1 conv summed-chan	1 conv aux mic
Training Condition	10 sec 2-chan	optional	optional	optional	optional
	1 conv 2-chan	optional	**required**	optional	optional
	3 conv 2-chan	optional	optional	optional	optional
	8 conv 2-chan	optional	optional	optional	optional
	3 conv summed-chan	optional	optional	optional	optional

Table 2: Enrollment and test conditions for SRE'05. The bold typed area is the required condition of the evaluation.

2.2.3 2006 SRE

In 2006 the conditions are almost same with SRE'05. The corpus for 2005 is reused and additional data was collected by the same way for 2006 too. 1conv-1conv condition is required to submit whereas other 14 conditions are optional. The required one and other possible tests are shown in Table 3. The test conditions of 2006 SRE are identical with those of 2005 so direct performance comparisons could be done fairly.

- Training Conditions

    1. A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)
    2. One two-channel conversation segment which has approximately five minutes total duration including silence
    3. Three two-channel conversation segment, each has approximately five minutes total duration including silence
    4. Eight two-channel conversation segment, each has approximately five minutes total duration including silence
    5. Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

- Test Conditions

    1. A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)

		Test Segment Condition			
		10 sec 2-chan	1 conv 2-chan	1 conv summed-chan	1 conv aux mic
Training Condition	10 sec 2-chan	optional			
	1 conv 2-chan	optional	**required**	optional	optional
	3 conv 2-chan	optional	optional	optional	optional
	8 conv 2-chan	optional	optional	optional	optional
	3 conv summed-chan		optional	optional	

Table 3: Enrollment and test conditions for SRE'06. The bold typed area is the required condition of the evaluation.

2. One two-channel conversation segment which has approximately five minutes total duration,i.e. with silence

3. A summed-channel conversations, created concatenating the two sides of existing conversations sample by sample.

4. One two-channel conversation segment where the usual telephone speech segment was replaced by microphone data for the target speaker side. This microphone data was provided with 8 kHz sampling rate and 8-bit $\mu - law$ encoding form.

2.2.4 2008 SRE

In 2008 short2-short3(1conv-1conv) condition was required out of 13 conditions. All possible training and test condition combinations are shown in Table 4. Unlike the past years interview data over microphone channel was added to the required trial segment both for training and test, additionally telephone data over microphone channel added to the test segment. Information of channel type, i.e. telephone or

microphone was given. Also it is known that whether the record is a usual telephone conversation or an interview scenario. Some of the data of 2006 SRE is reused for 2008 SRE.

Within the significant changes from the previous years conditions, it is possible to compare the results of conversational telephone data with 2006 SRE. For the rest of the conditions it is unfair.

- Training Conditions

    1. **10sec** : A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)
    2. **short2** : One two-channel conversation segment which has approximately five minutes total duration over telephone channel or three minutes over microphone channel within an interview scenario where mostly the target speaker speaks.
    3. **3conv** : Three two-channel conversation segment, each has approximately five minutes total duration including silence
    4. **8conv** : Eight two-channel conversation segment, each has approximately five minutes total duration including silence
    5. **long** : A single channel conversation segment which is eight minutes or more recorded over microphone channel, involving mostly the speech of the target speaker not the interviewer.
    6. **3summed** : Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

- Test Conditions

    1. **10sec** : A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)
    2. **short3** : One two-channel conversation segment which has approximately five minutes total duration over telephone channel or three minutes over microphone channel within an interview scenario where mostly the target speaker spoke.
    3. **long** : A single channel conversation segment which is eight minutes or more recorded over microphone channel, involving mostly the speech of the target speaker not the interviewer.
    4. **summed** : Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

2.2.5 2010 SRE

2010 SRE is similar to 2008 SRE except some minor changes on the core condition. Core condition unlike the previous years includes conversational telephone speech recorded over room microphone or conversation from an interview scenario additional to the conversational telephone data recorded over usual(wired or wireless) telephone channels. Some of the data from above mentioned all types is recorded by producing relatively high or low vocal effort. Instead of fixed length interview segments, varying length of interview segments are used. They vary from three minutes to fifteen minutes. Vocal effort information is not given whereas the knowledge of record type(i.e., telephone conversation or interview scenario, microphone or ordinary telephone channel) is given.

		Test Segment Condition			
		10sec	short3	long	summed
Training Condition	10sec	optional			
	short2	optional	**required**		optional
	3conv		optional		optional
	8conv	optional	optional		optional
	long		optional	optional	
	3summed		optional		optional

Table 4: Enrollment and test conditions for SRE'08. The bold typed area is the required condition of the evaluation.

Since the participants of previous years' evaluations were not so much interested in unsupervised adaptation mode and the performance measure was changed in 2010, unsupervised adaptation mode is not involved among the optional conditions any more. There are some changes in performance measurement parameters given by NIST. It will be given in details in Section 6.2.1. Since normal vocal effort English conversational telephone data is similar to those belonging to 2008 SRE, fair comparions can be done between two evaluations.

Out of 9 conditions shown in Table 5 core-core(1conv-1conv) test is the required one in SRE'10. The database consists of all English records and number of trials is increased a lot compared with previous years.

- Training Conditions

    1. **10sec** : A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture $\sim$10 seconds of speech.)

2. **core** : One two-channel conversation segment which has approximately five minutes total duration over telephone channel or three to fifteen minutes over microphone channel within an interview scenario where mostly the target speaker spoke.
3. **8conv** : Eight two-channel conversation segment, each has approximately five minutes total duration including silence
4. **8summed** : Eight summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

- Test Conditions

1. **10sec** : A sample from a two-channel conversation segment which has approximately 10 seconds speech for the target speaker(The NIST energy based automatic speech detector is used to capture ∼10 seconds of speech.)
2. **core** : One two-channel conversation segment which has approximately five minutes total duration over telephone channel or three minutes over microphone channel within an interview scenario where mostly the target speaker spoke.
3. **summed** : Three summed-channel conversations, created concatenating the two sides of existing conversations sample by sample. That conversations include a common speaker for all three summed and a second person who is different for each of them

2.2.6 2012 SRE

Most of the targets data are taken from the corpora used in the past evaluations by 2004 Large and varying number of sessions are used per target speaker. In SRE12

		Test Segment Condition		
		10sec	core	summed
Training Condition	10sec	optional		
	core	optional	**required**	optional
	8conv	optional	optional	optional
	8summed		optional	optional

Table 5: Enrollment and test conditions for SRE'10. The bold typed area is the required condition of the evaluation.

knowledge of all targets is allowed in computing each trials detection score. This differs from all previous SREs. Previously systems were restricted to use only knowledge of the single target speaker that was specified as the trial target.

There are 9 possible training and test segment combinations for 2012 SRE as can be seen in Table 6. Core-core test is the one that should be submitted.

The direct comparison of results partially possible between 2010 and 2012 because of the significant changes related with data amount, number of sessions, training and test conditions.

- Training Conditions

    1. **core** : All speech data available for each target speaker. There is no distinction according to data amount or channel type.
    2. **telephone** : All telephone channel data available for each target speaker. Microphone data from any target speaker can not be used, but microphone data belonging to others can be used for a background speaker purpose.
    3. **microphone** : All microphone channel data available for each target speaker. Telephone channel data from any target speaker can not be used, but telephone data belonging to others can be used for a background speaker

		Test Segment Condition				
		core	extended	summed	known	unknown
Training Condition	core	**required**	optional	optional	optional	optional
	microphone	optional	optional			
	telephone	optional	optional			

Table 6: Enrollment and test conditions for SRE'12. The bold typed area is the required condition of the evaluation.

purpose.

- Test Conditions

  1. **core** : A sample from a two-channel telephone conversation or interview segment which has between 20 and 160 seconds speech for the target speaker. Some of these utterances have additive noise.
  2. **extended** : Conditions are same as the conditions of **core** with larger number of trials than the **core** tests.
  3. **summed** : A summed-channel excerpt from a telephone conversation or an interview, created concatenating the two sides of existing conversations sample by sample. The excerpt has between 20 and 160 seconds of target speaker speech.
  4. **known** : The trial list used in **extended** condition is used here too. The system should presume that all of the impostors belong to known speakers.
  5. **unknown** : The trial list used in **extended** condition is used here too. The system should presume that all of the impostors belong to unknown speakers.

CHAPTER III

UNIVERSAL BACKGROUND MODEL (UBM)

State-of-the art speaker verification systems are mostly based on the *Gaussian mixture models* (GMM). GMM based approaches use a background model which is assumed to represent whole speaker space within all possible variabilities. Mathematically it is a large GMM trained using the available acoustic features of speakers. This speaker independent model is called *universal background model* (UBM).

UBM is utilized for a variety of purposes in speaker verification. It is used in classical *maximum a posteriori* (MAP) adaptation to train the speaker models [12]. In GMM-SVM approach GMM supervectors are used [11]. It is also important part of i-vector based systems, which is recently dominant method [4]. UBM is used to align the acoustic features before training the i-vector extractor which will be explained in Section 4.1.

In the decision part of a verification task, likelihood ratio test is mostly performed. It is a zero-one decision task consisting comparison of two hypothesis: H_0 for speaker model and H_1 for other speakers.

$$H_0 = O\ is\ from\ speaker\ S$$

$$H_1 = O\ is\ not\ from\ speaker\ S$$

As mentioned before, UBM is assumed to represent a large variety of speakers so it can be used as an alternative hypothesis for below purpose:

$$\Delta(X) = \frac{p(X|\lambda_S)}{p(X|\lambda_{\bar{S}})} \begin{cases} \geq \theta & accept H_0 \\ < \theta & reject H_0 \end{cases} \tag{1}$$

There is a myth which can be generalized as "'There is no data like more data"' for training UBM, so training dataset consists of tens of hours of speech data. There are some parameters that are related to algorithm and data separately while training UBM. Algorithm parameters are such as: proper mixture number, initialization method, training method, number of iterations. Which corpus to choose, number of speakers, data amount per speaker, selection way of features, variability that can be captured are the considered points related to data parameters. These concerns are in detail analyzed in [13]. It is shown that instead of gathering a large dataset and constructing UBM using whole data, smarter and faster algorithms can be utilized. Leading feature selection (LFS), uniform feature selection (UFS), random feature selection (RFS) and intelligent feature selection (IFS) from each utterance are the methods in detail worked on. The main point is using as less data as possible that satisfies enough variability and over-all performance for the system. As given in [12], over-all system performance saturates after some point while increasing the data amount. In parallel with this result, as the data amount raised the variability spanned also saturates.

3.1 GMM Training

In our work mixture splitting and random feature selection methods are used. Starting from two mixture, each mixture is split to two mixtures up to the desired mixture number reached. Defined percentage of data per speaker is randomly selected and used to train the GMM parameters.

One single Gaussian has the parameters mean (μ), variance (σ^2). Gaussian mixtures have parameter weights (w) in addition to those belonged to single Gaussian. The probability distribution function (pdf) of a GMM is defined as below.

$$p(x) = \sum_{m=1}^{M} w_m \mathcal{N}(x|\mu_m, \Sigma_m) \tag{2}$$

where the weight parameters w_m satisfy these two conditions: $0 \leq w_m \leq 1$ and $\sum_{m=1}^{M} w_m = 1$. To estimate the optimum parameters of GMM related with the training data, first likelihood function is defined. Since using logarithm of the likelihood function makes the computations easier(multiplications turn out to be summation and exponential term goes off), log-likelihood function of a GMM is defined instead of Eq. 2.

$$ln\ p(X|w,\mu,\Sigma) = \sum_{f=1}^{F} ln\left(\sum_{m=1}^{M} w_m \mathcal{N}(x_f|\mu_m,\Sigma_m)\right) \tag{3}$$

Optimum parameters that maximize the likelihood of data to the model are estimated using the expectation-maximization(EM) algorithm [14]. It is an iterative method since Eq. 3 has no closed-form solution that can be solved at one swoop.

1. GMM parameters; means μ_m, covariances Σ_m and mixture weights w_m are initialized and log likelihood is computed initially.

2. In **E step** the responsibilities of each mixture on the generation of each sample(frame) are calculated using the current parameter values of GMM.

$$\gamma(z_{fm}) = \frac{w_m \mathcal{N}(x_f|\mu_m,\Sigma_m)}{\sum_{j=1}^{M} w_j \mathcal{N}(x_f|\mu_j,\Sigma_j)} \tag{4}$$

3. In **M step** GMM parameters are estimated and updated to be used in the next iteration using the responsibilities evaluated above Eq. 4

$$\mu_m^{new} = \frac{1}{F_m}\sum_{f=1}^{F} \gamma(z_{fm}) x_f \tag{5}$$

$$\Sigma_m^{new} = \frac{1}{F_m}\sum_{f=1}^{F} \gamma(z_{fm})(x_f - \mu_m^{new})(x_f - \mu_m^{new})^T \tag{6}$$

$$w_m^{new} = \frac{F_m}{F} \tag{7}$$

where F_m, number of frames belonging to GMM component m is

$$F_m = \sum_{f=1}^{F} \gamma(z_{fm}) \tag{8}$$

4. The log likelihood function in Eq. 3 is computed and the convergence condition is examined to make a decision whether to keep iterating starting from step 2 or to stop.

CHAPTER IV

TOTAL VARIABILITY SPACE (TVS)

Total variability space (TVS) modeling which was introduced by Dehak [4] has recently been the mostly used paradigm in state of the art speaker verification systems. This model can be formulated as shown below

$$M_s = M_0 + Tw_s \tag{9}$$

In above model M_0 is the speaker independent supervector, namely concatenated UBM means. T is the low rank total variability matrix and called projection matrix or i-vector extractor too. w_s is the latent variable which is assumed to be normally distributed. Specifically w_s is called as i-vector. Using this model each utterance can be projected on to the space with a very low dimension compared with speaker supervector, M_s. TVS modeling both provides dimensionality reduction and channel compensation gathering whole variability in one space.

4.1 Training of The T-Matrix

In TVS modeling the eventual purpose is to extract a low dimensional i-vector for each speaker. For this purpose, firstly $T - matrix$, so called i-vector extractor should be trained using a large database consisting of speakers who have multiple sessions. Each session is assumed as a different speaker unlike the eigenvoice modeling in [15]. Training procedure has just this minor difference between eigenvoice modeling and TVS modeling.

Considering Eq. 9, it can be said that for any speaker s, frames aligned with k^{th} mixture of UBM, are distributed with mean $M_k(s)$ and covariance matrix Σ_k.

Σ denote the $DKxDK$ block diagonal matrix whose block diagonals are $\Sigma_1, ..., \Sigma_K$. Likelihood function over all speakers in the database is given as below

$$\prod_{s=1}^{S} \max_{w} P(\chi(s)|M_0 + Tw, \Sigma) \tag{10}$$

where $\chi(s)$ is speaker's training data and s ranges over all utterances in the training set. Since in total variability space both speaker and channel variabilities are considered, each utterance is assumed to belong to a different speaker.

4.1.1 EM Algorithm

Maximum likelihood (ML) estimation problem in Eq. 10 has no closed form solution so expectation maximization (EM) algorithm, an iterative method, is used. EM has two steps:

1) Current estimates of T and Σ are used to to find speaker supervector which maximizes the likelihood of each training data, $\chi(s)$ as shown below.

$$w(s) = \arg\max_{w} P(\chi(s)|M_0 + Tw, \Sigma) \tag{11}$$

2) V and Σ are updated by maximizing

$$\prod_{s=1}^{S} P(\chi(s)|M_0 + Tw_s, \Sigma) \tag{12}$$

In the E-step of EM algorithm, main computation is calculation of posterior distribution of $w(s)$ given the speaker's training data. This computation is done for all speakers and using current estimates (or in first iteration initial random values) of T and Σ.

For the calculation of posterior distribution purpose, each speaker's training data should be aligned with the speaker-independent model, UBM. Alignment means that each frame of the data is labeled by a mixture component. Using the alignment informations following statistics of $\chi(s)$ are extracted to be used in next calculations.

$N_k(s)$ is the number of frames of aligned with the k^{th} mixture, where $k = 1, ..., K$. First and zero order statistics $S_{X,k}(s)$ and $S_{XX^T,k}(s)$ are computed as follows:

$$S_{X,k}(s) = \sum_t (X_t - \mu_k) \tag{13}$$

$$S_{XX^T,k}(s) = \sum_t (X_t - \mu_k)(X_t - \mu_k)^T \tag{14}$$

where the summation is done over all frames X_t of speaker s aligned with the k^{th} mixture of speaker independent model UBM and μ_k is the k^{th} component of UBM mean vector.

$N(s)$ is defined as a $DKxDK$ block diagonal matrix whose block diagonals are $N_1(s)I, ..., N_K(s)I$. I, the identity matrix, is DxD matrix. $S_x(s)$ is defined as KD dimensional column vector, formed by concatenating $S_{X,1}, ..., S_{X,K}$ and $l(s)$ is defined as shown below:

$$l(s) = I + T^T \Sigma^{-1} N(s) T \tag{15}$$

After giving all necessary statistic definitions, expectation of posterior distribution of $w(s)$, $E[w(s)]$ and $E[w(s)w^T(s)]$ are given by these two formulas:

$$E[w(s)] = l^{-1}(s) T^T \Sigma^{-1} S_X(s) \tag{16}$$

$$E[w(s)w^T(s)] = E[w(s)]E[w^T(s)] + l^{-1}(s) \tag{17}$$

In the M-step, new model parameters T and Σ that maximize the Eq. 12 are calculated as below:

$$T^i \sum_{s=1}^{S} N_k(s) E[w(s)w^T(s)] = \sum_{s=1}^{S} S_X^i(s) E[w^T(s)] \tag{18}$$

$$\Sigma_k = \frac{1}{n_c}\left(\sum_{s=1}^{S} S_{XX^T,k}(s) - M_k\right) \tag{19}$$

Eq. 18 is just a linear equation system that is RxR. It is solved using basic linear algebra.

CHAPTER V

CHANNEL COMPENSATION ALGORITHMS

5.1 Linear discriminant analysis (LDA)

LDA is a classic dimensionality reduction technique that attempts to retain the dimensions that are most important for classification while removing the rest. To do that, LDA tries to maximize the between class covariance, S_b, and minimize the within class covariance, S_w, by maximizing the Rayleigh quotient in Eq. 47.

S_b and S_w are defined as follows:

$$S_b = \sum_{s=1}^{S} (\overline{w_s} - \overline{w})(\overline{w_s} - \overline{w})^T \tag{20}$$

$$S_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (w_i^s - \overline{w_s})(w_i^s - \overline{w_s})^T \tag{21}$$

where $\overline{w_s} = \frac{1}{n_s} \sum_{i=1}^{n_s} w_i^s$ is the mean of the i-vector for each speaker s, n_s is the number of sessions for speaker s, S is the total number of speakers, and $\overline{w}$ is the mean of all i-vectors and it is assumed to be null.

This maximization problem above, is solved by eigenvalue resolution. S_b and S_w are both symmetric matrices and their sizes are equal. The quotient in Eq. 47 which is desired to be maximized gets its largest value for eigenvector, v, related with the largest eigenvalue , λ, of matrix ${S_w}^{-1} S_b$. To maximize the quotient, the derivative with respect to v is taken and set to zero as below:

$\frac{(v^t S_w v)(2S_b v) - (v^t S_b v)(2S_w v)}{(v^t S_w v)^2} = 0$ which yields ${S_w}^{-1} S_b v = (\frac{v^t S_b v}{v^t S_w v}) v$.

Using the general eigenvalue equation:

$$S_b v = \lambda S_w v \tag{22}$$

a projection matrix $A = {S_w}^{-1}S_b$, which consists of the best eigenvectors (those with largest eigenvalues) in Eq. 22 of where λ is the diagonal matrix of eigenvalues. Dimensionality of the i-vectors are then reduced by multiplying them with the projection matrix A obtained as described above.

5.2 Probabilistic linear discriminant analysis (PLDA)

In [5], a probabilistic version of the LDA technique is proposed for the face recognition tasks. In [4], the PLDA algorithm was shown to be effective for reducing the intersession variability in speaker verification.

In the PLDA method, j^{th} utterance of i^{th} speaker, x_{ij}, is denoted as follows:

$$x_{ij} = \mu + Fh_i + Gw_{ij} + \epsilon_{ij} \tag{23}$$

where μ is the mean of all i-vectors in the training data.

Eq. 23 can be given as a linear system as below for a speaker with N sessions.

$$\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} \mu \\ \mu \\ . \\ . \\ . \\ \mu \end{bmatrix} + \begin{bmatrix} F & G & 0 & ... & 0 \\ F & 0 & G & ... & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ F & 0 & 0 & ... & G \end{bmatrix} \times \begin{bmatrix} h_1 \\ w_1 \\ w_2 \\ . \\ . \\ . \\ w_N \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ . \\ \epsilon_N \end{bmatrix}$$

or it can be summarized as below

$$x' = \mu' + Ay + \epsilon' \tag{24}$$

This probabilistic model consists of two parts: (i) the signal component $\mu + Fh_i$, which represents the speaker and (ii) the noise component $Gw_{ij} + \epsilon_{ij}$, which represents

the session variability given a speaker. F and G are factor loading matrices, h_i are the speaker factors, and w_{ij} are the channel factors. h_i and w_{ij} have Gaussian prior distributions, $\mathcal{N}(0; I)$. The residual noise ϵ_{ij} is defined to be Gaussian with a diagonal covariance matrix.

The parameters are $\theta = \{\mu, F, G, \Sigma\}$ are estimated with a Maximum Likelihood (ML) approach using the Expectation Maximization (EM) algorithm.

In the E-step posterior distribution over the latent variables, h_i and w_{ij} are calculated as below

$$E[y_i] = (A^T \Sigma'^{-1} A + I)^{-1} A^T \Sigma'^{-1} (x_i - \mu') \tag{25}$$

$$E[y_i y_i^T] = (A^T \Sigma'^{-1} A + I)^{-1} A^T \Sigma'^{-1} (x_i - \mu') \tag{26}$$

Then in M-step optimum parameter values are estimated as follows

rewriting the Eq. 23 as:

$$x_{ij} = \mu + \begin{bmatrix} F & G \end{bmatrix} \begin{bmatrix} h_i \\ w_{ij} \end{bmatrix} + \epsilon_{ij} \tag{27}$$

$$x_{ij} = \mu + B z_{ij} + \epsilon_{ij} \tag{28}$$

$$B = \left(\sum_{ij} (x_{ij} - \mu) E[z_i]^T \right) \left(\sum_{ij} E[z_i z_i^T] \right)^{-1} \tag{29}$$

$$\Sigma = \frac{1}{IJ} \sum_{i,j} \mathbf{Diag}[(x_{ij} - \mu)(x_{ij} - \mu)^T - B E[z_i](x_{ij} - \mu)^T] \tag{30}$$

5.3 Within-class covariance normalization (WCCN)

WCCN is used to whiten the within-class covariance matrix so that S_w becomes an identity matrix after normalization. Its asymptotic optimality properties have

been shown to improve the performance of SVM classifiers with generalized linear kernels [10]. When WCCN is used with CDS, the distance score is

$$score(w_1, w_2) = \frac{(B^T w_1)^T (B^T w_2)}{\|B^T w_1\| \|B^T w_2\|} \gtrless \theta \tag{31}$$

where B is obtained from the Cholesky decomposition of the inverse matrix of S_w, as $S_w^{-1} = BB^T$.

5.4 Nuisance attribute projection (NAP)

The NAP algorithm [11] attempts to remove the nuisance directions in the i-vector space that are not related to interspeaker variations. I-vectors are multiplied with a projection matrix $P = I - RR^T$ where R is a low rank matrix whose columns are the k eigenvectors corresponding to the k largest eigenvalues of the within class covariance matrix S_w in Eq. 21. Thus, the NAP technique confines the i-vectors in a lower dimensional space by removing the directions that cause most of the intersession variability. In that sense, it can be seen as an eigenchannel technique.

The new cosine distance after projection with P is

$$score(w_1, w_2) = \frac{(Pw_1)^T (Pw_2)}{\|Pw_1\| \|Pw_2\|} \gtrless \theta \tag{32}$$

CHAPTER VI

SCORING ALGORITHMS AND PERFORMANCE MEASUREMENTS

Once the speaker models are trained, next step is how to measure the success of these models. Verification scenario can be simplified as follows: there is a caller who claims to be a certain identity and the system is asked to make a decision either within a confidence value or not. In this context, frequently used algorithms are likelihood ratio (LR) test, cosine distance scoring (CDS), support vector machines (SVM). As mentioned in Section 7 generalized discriminant analysis (GDA) can also be counted in this group.

To examine the overall performance of a verification system, all trials are evaluated against their target speakers and the scores are obtained. The performance can be analyzed over the error rates or the costs predefined for each type of error. An automatic system may shorten the verification time or raise the security level in a call center scenario. The analysis results of this contribution will be given as an evaluation of performance too.

6.1 Scoring Algorithms

Scoring algorithms may primarily return a label that implies which class the caller belong to as in SVM. In CDS, the system computes the distance between the caller and the target speaker in the speaker space. Likelihood ratio test compares the likelihood of the target speaker model and the alternative model given the caller data.

6.1.1 Likelihood Ratio Test

There are two choices when a caller is needed to verify. It may be accepted that the caller and claimed ID are the same speaker or the claim is rejected. For these two options, two different hypothesis need to be formulated and tested. H_0 and H_1 are two hypothesis:

$$H_0 = w_c \; and \; w_t \; are \; from \; the \; same \; speaker$$

$$H_1 = w_c \; and \; w_t \; are \; from \; different \; speakers$$

where w_t is the i-vector of claimed(target) speaker and w_c for the caller.

$$\Delta(w) = \frac{p(w_t, w_c|H_0)}{p(w_t|H_1)p(w_c|H_1)} \begin{cases} \geq \theta & accept \; H_0 \\ <\theta & reject \; H_0 \end{cases} \tag{33}$$

Target speaker may have multiple sessions to be used for the enrollment. In that case test in Eq. 33 can be modified as follows:

$$\Delta(w) = \frac{p(w_{t1}, w_{t2}, ..., w_{tR}, w_c|H_0)}{p(w_{t1}, w_{t2}, ..., w_{tR}|H_1)p(w_c|H_1)} \begin{cases} \geq \theta & accept \; H_0 \\ <\theta & reject \; H_0 \end{cases} \tag{34}$$

where R denotes the number of sessions belong to speaker t.

6.1.2 Cosine distance scoring (CDS)

In cosine distance scoring (CDS), the cosine distance between the claimed speaker's i-vector, $w_{claimed}$, and the test speaker's i-vector, w_{test}, is calculated as follows:

$$score(w_{claimed}, w_{test}) = \frac{w_{claimed}^T w_{test}}{\|w_{claimed}\| \|w_{test}\|} \tag{35}$$

Then, $score(w_{claimed}, w_{test})$ is compared with an hard-threshold θ to make the verification decision. CDS is a computationally simple method and its performance

is often comparable to the SVM and GDA methods.

6.1.3 Support vector machines (SVM)

Support vector machine (SVM) [9] is a popular supervised binary classifier. Similar to GDA, it works in a high dimensional, or infinite dimensional, space by using the kernel trick. In training, supervised data is given as

$$D = \{(x_i, y_i)|x_i \in \mathbf{R^N}, y_i \in \{-1, +1\}\},\ i = 1, .., M \tag{36}$$

where x_i is an N-dimensional supervector, y_i is the classification output which is either +1 or -1, and M is the total number of training samples. The aim of SVM is to find a hyperplane between the two classes such that the margin between them is maximum which is achieved when the classes are separable. The decision function is defined as follows:

$$f(x) = w^T\phi(x) + b = \sum_{i=i}^{N} y_i\beta_i k(x, x_i) + b \tag{37}$$

where x is the input vector, x_i are the support vectors and N is the total number of support vectors. Support vectors, w, β_i and b parameters are calculated in the training phase. The final classification decision for the input vector x is the sign of $f(x)$.

6.2 Performance Measurements

6.2.1 Error Rates and Costs

Output of a speaker verification system can be a zero-one decision(hard thresholding) or a soft score describing the reliability level. Considering the two type of trials: client(target speaker) and impostor, performance measurements can be done. Client is the trial where claimed speaker and actual speaker are same on the other hand impostor is the trial where claimed speaker and actual speaker are different.

The success of any system is mostly measured by its error rate and efforts are based on the degradation this rate. In speaker verification there are two type of error rates related with these two trials: false alarm probability and miss probability. False alarm probability, P_{FA}, is the rate of impostors who are accepted as target speakers. Miss probability, P_M, is the rate of target speakers who are rejected by the system being considered as betrayers.

Equally error rate (EER) and detection cost function (DCF) are mostly used metrics in literature. These metrics are defined by National Institutes of Standards and Technology(NIST). EER is the point where the P_{FA} and P_M are equal. There is a trade-off between P_{FA} and P_M within varying threshold in decision phase as shown in Eq. 1. This trade-off can be figured using Detection Error Trade-off (DET) curve so that performance and calibration of the system can be visually analyzed. An example of a DET curve can be seen in Figure 3. The tool used to figure it out is provided by NIST. Actually it computes error rates for each threshold changing by an epsilon within a range.

Detection cost function (DCF) is the fundamental metric utilized in the Speaker Recognition Evaluations (SREs) conducted by NIST. In DCF instead of searching the point where the error rates are equal, weighted sum of P_{FA} and P_M are considered. DCF is defined as below[16]:

$$C_{Det} = C_M \times P_{M|Target} \times P_{Target} + C_{FA} \times P_{FA|Impostor} \times P_{Impostor} \tag{38}$$

where C_M and C_{FA} are relative costs of rejection of a target speaker and false acceptance of a non-target speaker. P_{Target} is the prior probability specified for the target speaker, it sums up to 1 with $P_{Impostor}$. The parameter values in Table 7 were used as the primary evaluation metric of the verification performance in all speaker detection tasks defined by NIST in 2004, 2005, 2006 and 2008.

By 2010 SRE, NIST had made a change in cost model parameters. For core and

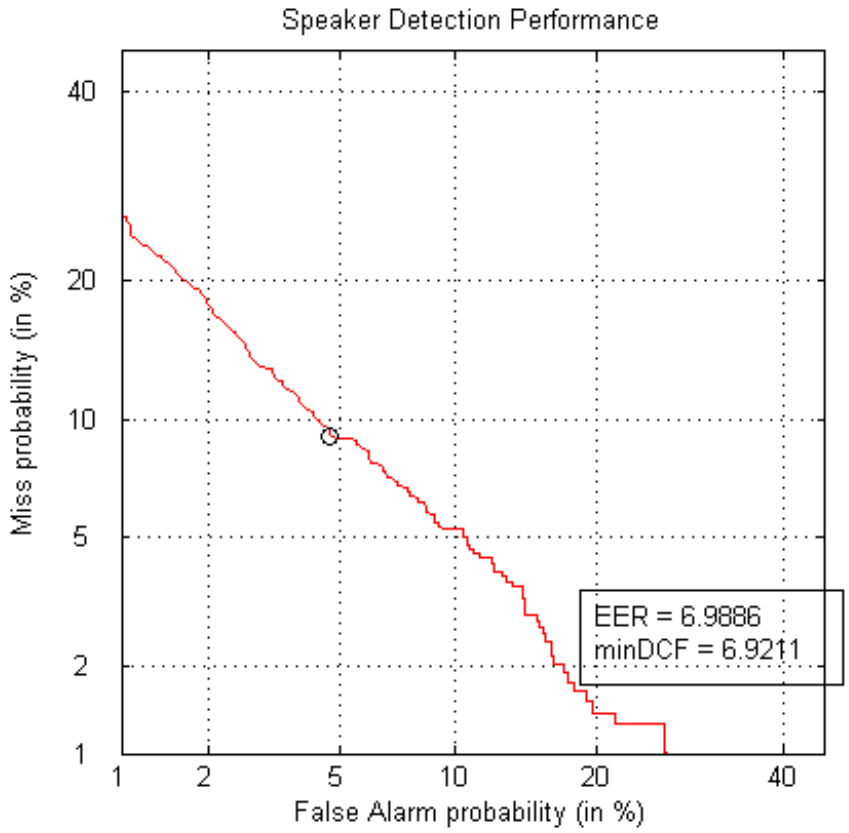

Figure 3: A DET curve sample. It depicts the trade-off between two error rates: false alarm and miss. The smoother the slope the better the calibration is said to be obtained.

8conv-core test conditions, the weight of false alarm probabilities were increased by decreasing the cost of miss, C_M, and prior probability of being target, P_{Target} as can be seen with comparison of Table 7 and Table 8. However for the rest of the test conditions of SRE 2010, parameters in Table 7 were still primarily used while those parameters were used in the evaluation of all conditions

To improve the intuitive meaning of C_{Det} calculated in Eq. 38, it is normalized dividing by the best cost that could be obtained as below:

$$C_{Default} = min\{C_M \times P_{Target},\ C_{FA} \times P_{Impostor}\} \tag{39}$$

$$C_{Norm} = C_{Det}/C_{Default} \tag{40}$$

$\mathbf{C_M}$	$\mathbf{C_{FA}}$	$\mathbf{P_{Target}}$
10	1	0.01

Table 7: Speaker Detection Cost Model Parameters for the primary evaluation decision strategy in NIST SREs

$\mathbf{C_M}$	$\mathbf{C_{FA}}$	$\mathbf{P_{Target}}$
1	1	0.01^4

Table 8: Speaker Detection Cost Model Parameters for the core and 8conv-core test segment conditions in 2010 SRE

For SRE12 detection cost was still basic performance metric, but there were two important changes done related with the computation of this cost compared with the previous evaluation SRE10 [17]:

1. Trial scores should be submitted as log likelihood ratios so the sides did not need to submit any decision output. Detection decisions were determined by thresholding the log likelihood scores using the threshold obtained from cost parameters where threshold is a known function of them.

2. The primary cost measure in SRE12 was a combination of two costs where SRE10 parameters and a greater target prior were used together. The purpose of this change was to add to the stability of the cost calculation and to increase the significance of score calibration over a wide range of likelihood ratios.

The cost function used in SRE12 for computation of costs for known and unknown impostors (non-target speakers).

$$\begin{aligned} C_{Det} =& C_M \times P_{M|Target} \times P_{Target} + C_{FA} \times (P_{FA|KnownNonTarget} \times P_{Known} \\ &+ P_{FA|UnknownNonTarget} \times (1 - P_{Known})) \times P_{Impostor} \end{aligned} \tag{41}$$

		C_M	C_{FA}	$P_{Target-A1}$	$P_{Target-A2}$	P_{Known}
Test Segment Condition	extended summed core	1	1	0.01	0.001	0.5
	known	1	1	0.01	0.001	1
	unknown	1	1	0.01	0.001	0

Table 9: Speaker Detection Cost Model Parameters for all test segment conditions in 2012 SRE

where P_{Target} is the prior probability that the trial speaker is the target speaker, P_{Known} is the prior probability that the impostor is one of the target speakers in the evaluation set. Detection cost model parameters are given in Table 9.

$$C_{Default} = C_M \times P_{Target} \tag{42}$$

$$\begin{aligned} C_{Norm} = P_{Miss|Target} + P_{Known} \times P_{FA|KnownTarget} \\ +\beta \times (1 - P_{Known} \times P_{FA}|UnknownNonTarget) \end{aligned} \tag{43}$$

where $\beta = \frac{C_{FA}}{C_M} \frac{1-P_{Target}}{P_{Target}}$

Actual detection costs was computed from the trial scores applying the thresholds of $log(\beta)$ for the two values of β, with β_{A1}(for $P_{Target-A1}$) being 99 and β_{A2}(for $P_{Target-A2}$) being 999.

The primary cost measure for SRE12 was defined as:

$$C_{primary} = \frac{C_{Norm}\beta_{A1} + C_{Norm}\beta_{A2}}{2} \tag{44}$$

6.2.2 Semi-automatic Approach to Verification

In some applications, such as customer verification in the call centers of banks, probability of false alarm (P_{FA}) has a very high cost and the current fully-automatic

systems cannot guarantee low enough P_{FA}. In the semi-automatic approach, we assume the presence of a human agent that asks security questions to the clients in addition to the automatic verification system. This scenario can be illustrated as in Figure 4[1] [18]. The role of the automatic system here can be two folds: it can either reduce the number of questions that the agent has to ask or it can improve the safety of the system. Our focus in this paper is on the reduction of questions which reduces the load on the human agent.

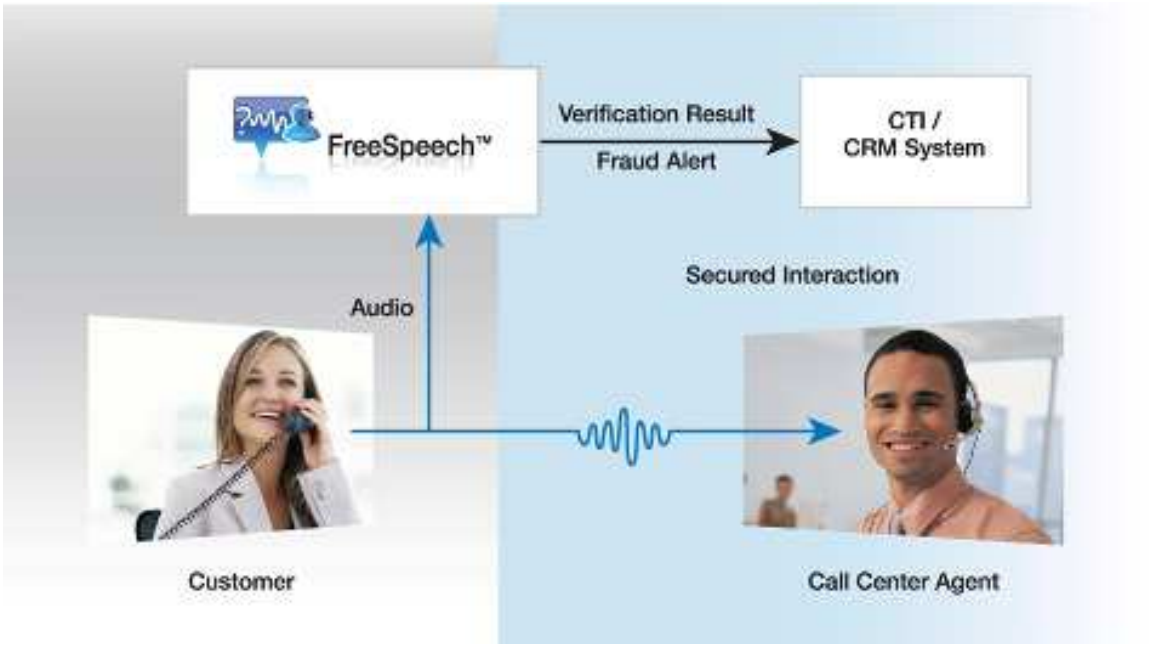

Figure 4: Automatic verification system aided call center scenario. The agent asks certain number of questions according to the score returned by the system. No additive effort needed for the agent, scoring is done during the flow of natural conversation.

Because a semi-automatic system is proposed here with the goal of reducing the number of security questions, classic performance metrics such as EER is not suitable for measuring the performance. Instead, in accord with the task, the performance metric used here is the reduction in the number of questions a human agent has to ask a client without an increase in P_{FA} compared to a fully-manual system.

[1]CTI:computer telephony integration, CRM:customer relationship management

In a fully-manual system the agent asks N_q questions to all clients. A binomial distribution is used to model the P_{FA} given that a question is answered correctly. Each question is assumed to have identical and independent distribution. If Q security questions are asked and all of them are answered correctly by the caller, then

$$P_{FA} = (1/k)^Q \tag{45}$$

where addition of every question increases P_{FA} by factor of k. Similarly, elimination of J security questions increases P_{FA} by a factor of $(1/k)^J$. To compensate, the automatic system should have a $P_{FA} \leq (1/k)^J$. Assuming that the automatic verification score is independent of the questions, the resulting semi-automatic system has a new $P_{FA}^{new} \leq P_{FA}$ which guarantees that the semi-automatic system is at least as good as the fully-manual system.

As opposed to the binary decisions of most of the current systems, the proposed systems here have N_q decision classes. If the system decision is class-i (C_i), where $i \in 1, 2..., N_q$, then the agent asks the caller i questions.

All of our systems are tested and compared using this new metric. Moreover, classifier fusion is used at the score and decision levels in an attempt get further improvements in performance. Details of those approaches are given below.

6.2.2.1 Single-Classifier Comparison

In this approach, each classifier has $(N_q - 1)$ number of threshold levels, γ_j, where $\gamma_{j-1} < \gamma_j$. The system decides on C_i, if $\gamma_{i-1} \leq S_i \leq \gamma_i$ where S_i the soft score of the classifier. The threshold levels are tuned independently for each classifier such that P_{FA} given C_i, P_{FA/C_i}, is less than or equal to $(1/k)^{i-1}$ with the objective that the thresholds are set as high as possible. Such an approach has the benefit of minimizing the number of questions without any sacrifice from the safety of the system.

6.2.2.2 Multiple-Classifier Fusion

Besides comparing the individual systems, experiments with the score fusion techniques have also been done to boost the performance. Logistic regression is used to fuse the scores of the three systems. Before fusing the system scores, all scores are first transformed into approximate probabilities using logistic regression. This step is performed for normalization purposes. NIST 2006 database is used for training the logistic regression parameters. Focal tool is used for logistic regression [19]. Once the scores are fused, similar to the individual classifier case, 5-fold cross-validation is used to measure the performance for multiple-classifier fusion on the 2008 test data.

A second approach commonly used in multi-classifier systems is decision fusion. A majority voting scheme was used in which the final decision is the class that gets the majority of the votes from the individual classifiers. If none of the classes get the majority of the votes, median number of questions suggested by the individual classifiers is used by the human agent. Some other alternative techniques were also tried such as using the minimum or maximum number of questions suggested by the classifiers. Because there was no significant gain in performance compared to individual systems the result for above fusion techniques are not reported here.

CHAPTER VII

GENERALIZED DISCRIMINANT ANALYSIS (GDA)

Real life data is often not linearly separable and hence linear classification techniques such as linear discriminant analysis (LDA) are not adequate in many situations. In such cases, kernel methods can be powerful in discriminating between data classes. In the kernel approach, i-vectors can be mapped to a higher dimensional space $\mathcal{F}$ with a mapping function $\phi(w)$. Instead of actually computing $\phi(w)$, the kernel trick

$$k(w_i, w_j) =< \phi(w_i), \phi(w_j) > \tag{46}$$

is used where k is the kernel function.

Similar to LDA, its kernel variant, GDA aims to maximize the Rayleigh quotient

$$J(v) = \frac{v^t S_b v}{v^t S_w v} \tag{47}$$

but it operates on $\phi(w) \in \mathcal{F}$ vectors instead of the i-vectors w. The kernel trick is not directly usable in the GDA case. To express the Rayleigh quotient with the kernel function, the following transformation is used:

$$v = \sum_{i=1}^{S} \sum_{j=1}^{n_i} \alpha_{ij} \phi(w_{ij}) \tag{48}$$

where v is a discriminant direction in the space spanned by $\phi(w_{ij})$. The objective function J in Eq. 47, which will be given in Section 7.1 becomes

$$J(\alpha) = \frac{\alpha^T K D K \alpha}{\alpha^T K K \alpha} \tag{49}$$

where K is an nxn kernel matrix defined as $K_{ij} = k(w_i, w_j)$ for $i = 1, ..., n; j = 1, ..., n$ where n is the total number of the training data for all speakers and all sessions.

α is a vector with component α_i. Note that if more than one v is used (multi-class GDA), α_i from all vectors v are concatenated. $D = diag(D_1, ..., D_s)$ where s is the number of classes(speakers), D_i is a diagonal matrix with all elements equal to $1/n_i$ where n_i is the number of samples in class i. In the tests, both with one-versus-all training (binary classification) and multi-class training where each class corresponds to a speaker were experimented.

In training GDA, maximization of J is solved in terms of α which lands itself to a generalized eigenvalue problem similar to LDA as described in more detail in Section 5.1. In testing, projection of a test point x onto v can again be represented using α and the kernel function by

$$v^T \phi(x) = \sum_{i=1}^{n} \alpha_i k(w_i, x) \tag{50}$$

If GDA is used in one-versus-all scenario, the 1-d score calculated in Eq. 50 is directly used for verification. For multi-class GDA, a vector is returned for each test data. Cosine distance scoring is then used to calculate the final score.

7.1 Rayleigh Quotient in kernel approach

As mentioned before in GDA instead of i-vector, w, $\phi(w)$ is used in space $\mathcal{F}$. Total covariance of i-vectors into $\mathcal{F}$ can be denoted as below:

$$S_w = \frac{1}{n} \sum_{i=1}^{S} \sum_{j=1}^{n_s} \phi(w_{ij}) \phi^t(w_{ij}) \tag{51}$$

Eigenvalue resolution, finding the eigenvalues λ and eigenvectors v that are the solutions of this equation is as follows:

$$\lambda S_w v = S_b v \tag{52}$$

To derive the Eq. 49, Eq. 52 is multiplied by $\phi^t(w_{si})$ and resulting equation:

$$\lambda\phi^t(w_{si})S_w v = \phi^t(w_{si})S_b v \tag{53}$$

has the same eigenvectors as Eq. 52 [20].

The left term of Eq. 53 yields using Eq. 48 :

$$\begin{aligned} S_w v = & \quad \frac{1}{n}\sum_{k=1}^{S}\sum_{l=1}^{n_k}\phi(w_{kl})\phi^t(w_{kl})\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\phi(w_{mp}) \\ = & \quad \frac{1}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\sum_{k=1}^{S}\sum_{l=1}^{n_k}\phi(w_{kl})[\phi^t(w_{kl})\phi(w_{mp})] \end{aligned}$$

$$\begin{aligned} \lambda\phi^t(w_{si})S_w v = & \quad \frac{\lambda}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{kl}\phi^t(w_{si})\sum_{k=1}^{S}\sum_{l=1}^{n_k}\phi(w_{kl})[\phi^t(w_{kl})\phi(w_{mp})] \\ = & \quad \frac{\lambda}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\sum_{k=1}^{S}\sum_{l=1}^{n_k}[\phi^t(w_{si})\phi(w_{kl})][\phi^t(w_{kl})\phi(w_{mp})] \end{aligned}$$

Using this formula for all speaker s and for its sessions i:

$$\lambda(\phi^t(w_{si}),...,\phi^t(w_{si}),...,\phi^t(w_{si}),...,\phi^t(w_{si}))S_w v = \frac{\lambda}{n}KK\alpha \tag{54}$$

The right term of Eq. 53 yields this:

$$\begin{aligned} S_b v = & \quad \frac{1}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\phi(w_{mp})\sum_{k=1}^{S}n_k[\frac{1}{n_k}\sum_{l=1}^{n_k}\phi^t(w_{kl})][\frac{1}{n_k}\sum_{l=1}^{n_k}\phi^t(w_{kl})]^t \\ = & \quad \frac{1}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\sum_{k=1}^{S}[\sum_{l=1}^{n_k}\phi(w_{kl})][\frac{1}{n_k}][\sum_{l=1}^{n_k}\phi^t(w_{kl})\phi^t(w_{mp})] \\ \phi^t(w_{si})S_b v = & \quad \frac{1}{n}\sum_{m=1}^{S}\sum_{p=1}^{n_m}\alpha_{mp}\sum_{k=1}^{S}[\sum_{l=1}^{n_k}\phi^t(w_{si})\phi(w_{kl})][\frac{1}{n_k}][\sum_{l=1}^{n_k}\phi^t(w_{kl})\phi(w_{mp})] \end{aligned}$$

Using this formula for all speaker s and for its sessions i:

$$(\phi^t(w_{11}),...,\phi^t(w_{1n_1}),...,\phi^t(w_{si}),...,\phi^t(w_{S1}),...,\phi^t(w_{Sn_S}))S_b v = \frac{1}{n}KDK\alpha \tag{55}$$

After gathering Eq. 54 and Eq. 55 and multiplying by α^t, $\lambda KK\alpha = KDK\alpha$ and $\lambda = \frac{\alpha^t KK\alpha}{\alpha^t KDK\alpha}$ are obtained respectively.

CHAPTER VIII

EXPERIMENTS

In speaker recognition community NIST SREs play a significant and dominant role guiding the direction of researches. Here NIST as an organizer gets the suggestions of participants and defines the problems and new challenges for the following years. As a new group in the community we have participated in both NIST SRE 2010 and SRE 2012. Sure it was not that easy to catch up with the community. Not only the algorithms to be implemented also understanding the way that SREs work, getting the database to be ready to use, text processing were all time consuming issues related with preparation period.

In this chapter, details about the database and algorithms used for the verification system will appear part by part. Our current system is an i-vector based system. The details of dataset organization is given first. Sequentially front-end part, learning part and testing part will be given.

8.1 Dataset Organization

8.1.1 Front-End

In front-end part relevant features of raw speech are extracted. Features used in a speaker verification system can be classified as high-level features, prosodic&spectrotemporal features and short-term spectral and voice source features [1] as shown below Figure 5.

High-level features are behavioral things that depend on socio-economic status, education, born place, language background, personality, environment where grown up. On the other hand low-level features are physiological things affected by size of vocal folds, length and dimension of the vocal tract. A brief comparison of high and low level features are shown in Figure 6.

1. High-level features
    - Phones, idiolect(personal lexicon), semantics, accent, pronunciation are in these type of features.
2. Prosodic&spectro-temporal features
    - Pitch, energy, duration, rhythm, temporal features are in these group
3. Short-term spectral and voice source features(Low-level features)
    - Spectrum and glottal pulse features are low-level features.

Figure 5: Feature categories and sample of each category

In text-independent speaker verification systems mostly low-level features are used and they do sufficiently good job. As can be seen from the Figure 6, low-level features are more suitable to process a huge database without text information and independent from language like NIST provides getting a good performance.

In our experiments mel frequency cepstral coefficients are extracted. 12 mel frequency cepstral coefficients (MFCC) with their delta coefficients and delta coefficient of log energy are used making 25 dimensional feature vectors. 20 ms Hamming window with shift of 10 ms is used in analysis. Energy-based voice activity detection (VAD) is used where silence and speech are classified using the bimodal histogram method.

8.1.2 Training The Voice Signatures

In GMM based approaches, firstly universal background model was trained using the extracted features of training dataset. That model fits on the specified number of mixtures of Gaussians. Here 512 component GMM was used for UBM with a diagonal covariance matrix. In Table 10 it can be seen that, for training UBM, switchboard(Switchboard-1 Release 2)[21], NIST SRE 2004 and SRE 2005 databases were used. Random sampling and mixture splitting methods were used while training UBM. Random sampling was applied randomly choosing 10% of each utterance from

Pros&Cons(High-level features)

+ Robust against channel effects and noise
- Difficult to extract
- A lot of training data needed
- Delayed decision making

Pros&Cons(Low-level features)

+ Easy to extract
+ Small amount of data is enough
+ Text and language independent
+ Real time recognition
- Sensitive to noise and mismatch

Figure 6: Comparison of high-level and low-level features in terms of advantages and disadvantages

the above given database. Iteration number was chosen 25 for expectation maximization (EM) algorithm used in maximum likelihood (ML) estimation.

Total variability space matrix, T, (also named i-vector extractor) was trained on those same databases with UBM. Since T is a low rank matrix, dimension of i-vectors were chosen 400 which is mostly like that in literature. Iteration number was 5 for training the i-vector extractor. 400 dimensional i-vectors were extracted as a signature for each speaker. For conditions that speakers have multiple sessions, in CDS, all sessions of speakers were concatenated and one single i-vector for each speaker was obtained.

8.2 Experiment Setup

Experiments are performed on the NIST 2006 and 2008 SRE database. One conversation and eight conversation sides are used in enrollment. Both core condition (5 minute) and 10 seconds test data are used for testing. Experiments are performed

	Switchboard	NIST 2004	NIST 2005
UBM	x	x	x
T	x	x	x
PLDA		x	x
WCCN		x	x
NAP		x	x
GDA		x	x

Table 10: Database usage organization for different training purposes.

for male speakers and telephone speech (landline and cellphone). Equal-error-rate (EER) is reported for all conditions as the evaluation metric.

CDS, SVM, and GDA algorithms described in Section 6 are used as scoring algorithms. WCCN, PLDA, NAP, and multi-class GDA algorithms described in Section 5 are used for channel compensation. Final verification decision is done by hard-thresholding on the soft scores as discussed in Section 6.2.1.

UBM was trained on telephone data from the NIST 2004, 2005 SRE and Switchboard Cellular-1 databases. Total variability space matrix, T, was trained on those same databases. WCCN, PLDA, NAP, and GDA are trained on the NIST 2004 and 2005 SRE databases. A table that summarizes the databases used for training the systems are shown in Table 10.

400 dimensional i-vectors are used for all systems. In the SVM-based approach, to address the scarcity of client data in training, penalized cost function is used to put high weight on missed detections. NAP is used for channel compensation for SVM with a rank of 150. With the PLDA system, dimensionality of the i-vectors is reduced to 300 while the number of channel factors is 100. Instead of a likelihood based approach to scoring, cosine distance scoring (CDS) was used after dimensionality reduction because of the substantial gain in computational complexity with CDS. Similarly, CDS was used when dimensionality is reduced with PLDA.

GDA algorithm is used both in one-versus-all and multi-class configurations. One-versus-all GDA is used with linear, polynomial, Gaussian, and Radial Basis Function

(RBF) kernels and the best results are obtained with the polynomial kernel. Therefore, the results with the polynomial kernel are reported here for the one-versus-all configuration. For the multi-class configuration, RBF kernel consistently outperformed the other kernels. Therefore, RBF kernel is used for the multi-class GDA tests. For multi-class GDA, dimensionality of the i-vector is reduced to 300. Then, those lower dimensional vectors are scored with CDS. Tuning of the kernel type and kernel parameters is done on the NIST 2008 SRE database.

8.2.1 Performance of the baseline systems

8.2.1.1 Results

In Table 11, our results with NIST 2006 and 2008 SRE are presented for 5 minute and 5 minute test data cases. GDA classifier with WCCN channel compensation scheme consistently outperformed all other systems. Channel compensation methods do not seem to have much effect on SVM probably because SVM is used with the NAP algorithm that already compensates for the channel effects.

	SRE 2006		**SRE 2008**	
	no WCCN	**WCCN**	**no WCCN**	**WCCN**
TVS	3,68	2,64	5,5	4,33
PLDA	3,3	2,53	4,53	5,04
SVM	5	3,46	5,71	4,53
SVM+NAP	3,84	4,37	6,08	5,37
GDA multiclass	2,67	2,78	3,81	3,81
GDA	3,57	**2,39**	4,31	**2,66**

Table 11: Comparison of all individual techniques with or without WCCN. Results are given on NIST 2006 and 2008 SRE databases for 1conv1conv condition. Performance metric is EER (in%).

Experiment results for 5 minute and 10 sec test data cases with NIST 2006 and 2008 SRE are shown in Table 12. PLDA with WCCN channel compensation scheme outperformed all other systems for NIST 2006 whereas GDA multiclass classifier outperformed all others for NIST 2008.

In Table 13, our results with NIST 2006 and 2008 SRE are presented for 8 session

	SRE 2006		SRE 2008	
	no WCCN	**WCCN**	**no WCCN**	**WCCN**
TVS	8,94	8,46	11,92	12,2
PLDA	8,2	**7,52**	10,4	12,23
SVM	11,4	10,01	15,18	13,08
SVM+NAP	10,73	11,71	14,47	15,13
GDA multiclass	8,9	8,27	11,03	**10,3**
GDA	9,32	8,52	13,34	10,5

Table 12: Comparison of all individual techniques with or without WCCN. Results are given on NIST 2006 and 2008 SRE databases for 1conv10sec condition. Performance metric is EER (in%).

enrollment where each of them is 5 minute and 10 sec test data cases. GDA classifier outperformed the other systems as expected when used with channel compensation, WCCN for NIST 2006. On the other hand for NIST 2008 data, PLDA outperformed the others even without using the channel compensation technique, WCCN. This is inconsistent with the above results.

	SRE 2006		SRE 2008	
	no WCCN	**WCCN**	**no WCCN**	**WCCN**
TVS	8,81	9,39	4,3	3,88
PLDA	8,38	9,54	**2,38**	3,37
SVM	9,17	8,13	4,25	5,88
SVM+NAP	10,99	10,53	7,36	6,93
GDA multiclass	9,38	8,2	8,82	6,86
GDA	8,88	**7,6**	5,31	3,19

Table 13: Comparison of all individual techniques with or without WCCN. Results are given on NIST 2006 and 2008 SRE databases for 8conv10sec condition. Performance metric is EER (in%).

In Table 14, our results with NIST 2006 and 2008 SRE are presented for 8conversations 5 minute and 10 seconds test data cases. These 10sec excerpts are obtained from 5min length records(8conv1conv) in order to obtain exactly fixed 10sec speech data. GDA classifier using the channel compensation, WCCN again outperformed the others for NIST 2006 data. But for NIST 2008 data, although GDA did a good job, SVM classifier outperformed all others when used within WCCN.

Multi-class GDA-based dimensionality reduction with CDS classification results

	SRE 2006		SRE 2008	
	no WCCN	**WCCN**	**no WCCN**	**WCCN**
TVS	4,64	5,45	3,35	3,27
PLDA	4,48	5,87	2,28	4,86
SVM	4,47	4,36	2,27	**1,77**
SVM+NAP	5,81	5,76	3,26	2,8
GDA multiclass	5,07	4,77	4,57	5,61
GDA	4,19	**3,55**	2,85	2,25

Table 14: Comparison of all individual techniques with or without WCCN. Results are given on NIST 2006 and 2008 SRE databases for 8conv1conv-10sec condition. Performance metric is EER (in%).

are shown for each test condition. Performance with multi-class GDA is significantly worse than the one-versus-all results. Thus, the multi-class case was not investigated any further.

8.2.2 Semi-automatic verification experiments

Semi-automatic verification experiments are done using the NIST 2006 and 2008 SRE database with 8conv training data both for 5min and 10sec test conditions. These 10sec excerpts are obtained from 5min length records(8conv1conv) in order to obtain exactly fixed 10sec speech data. The 8conv10sec condition results for 2006 SRE client and impostor trials are shown in Figure 7 and Figure 8 respectively. The first clear result from this figure is that most of the gain is obtained with the client data as expected. Reductions in the impostor data is quiet modest compared to the client data. Another observation from Figure 7 and Figure 8 is that the SVM system seems to perform poorly compared to GDA and PLDA while the GDA system seems to outperform the other two systems in most cases.

In the next phase, experiments have been done with the same test condition for 2008 SRE. The results are shown in Figure 9 and Figure 10. In this case, all algorithms perform similarly for the clients and the differences are not significant. For the impostors, there are insignificant differences between the algorithms that change with k.

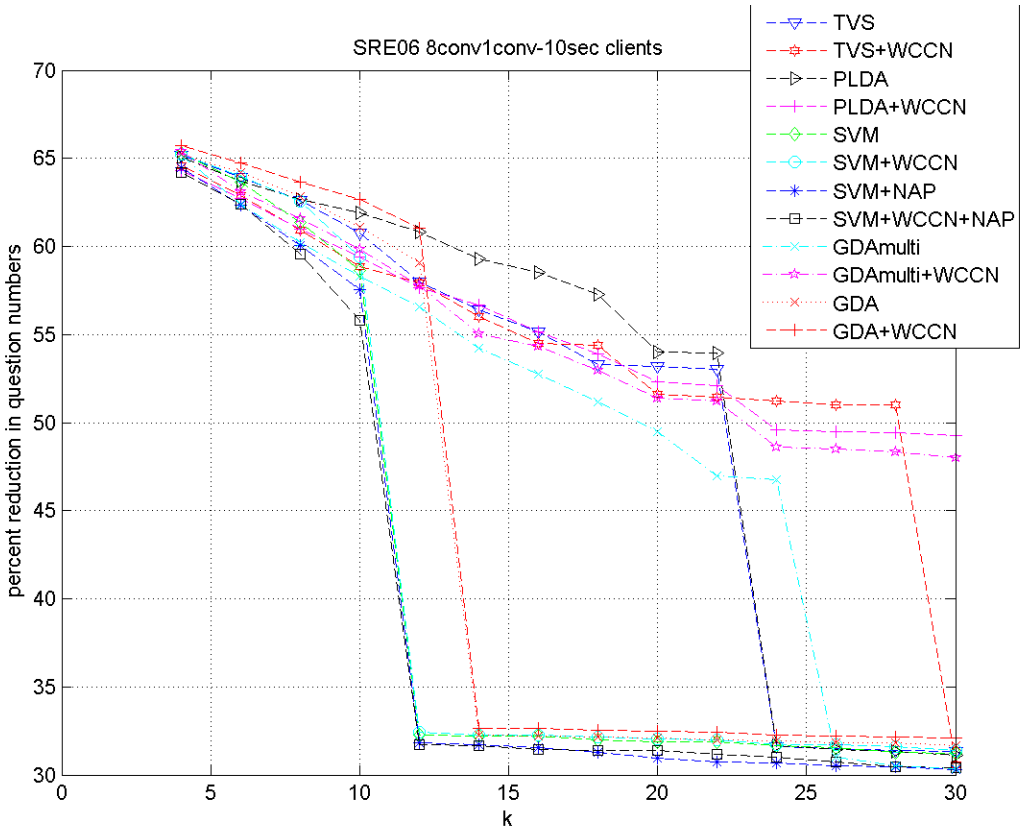

Figure 7: Reduction in number of questions with varying k values. Results are given on NIST 2006 SRE, 8conv1conv-10sec test setup.

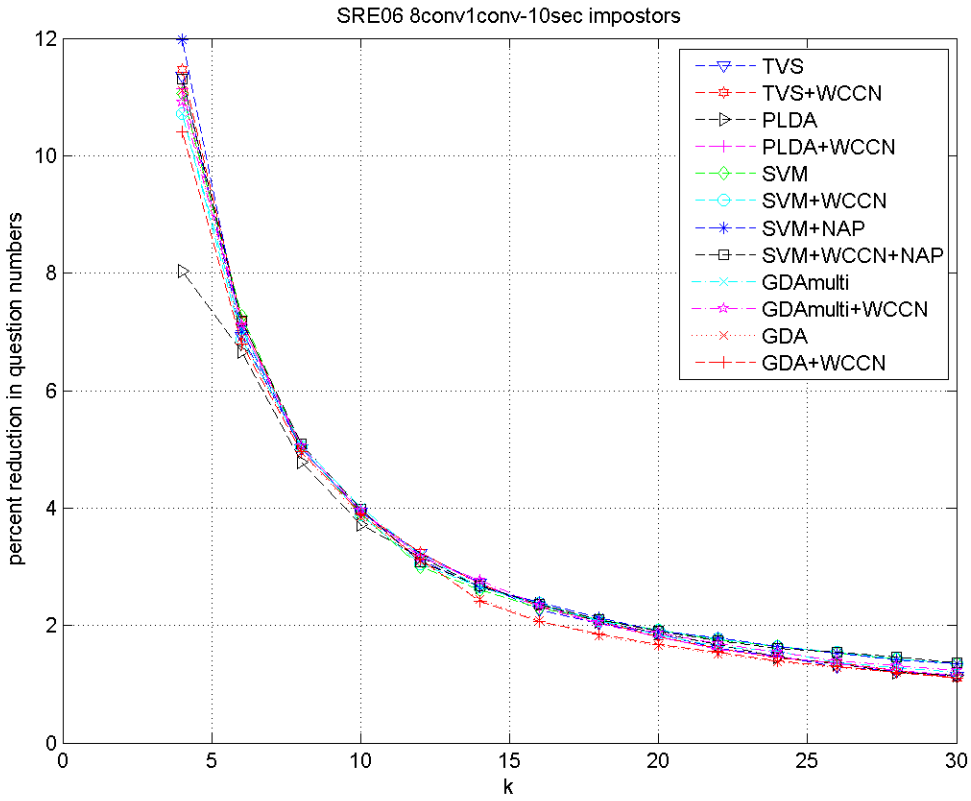

Figure 8: Reduction in number of questions with varying k values. Results are given on NIST 2006 SRE, 8conv1conv-10sec test setup.

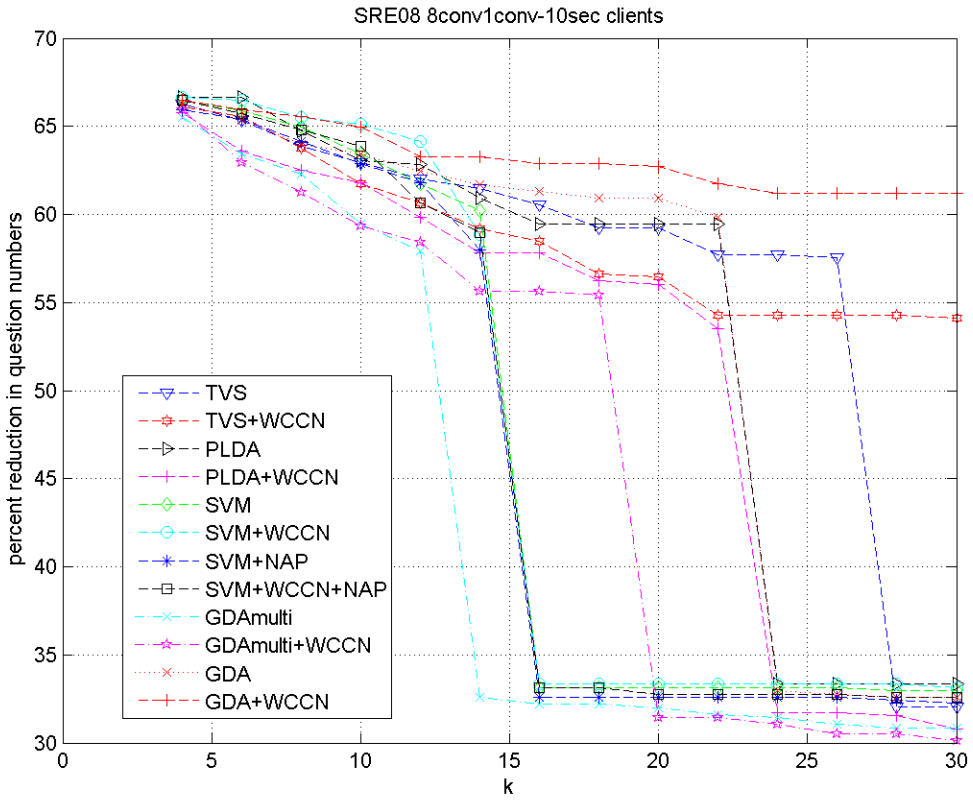

Figure 9: Reduction in number of questions with varying k values. Results are given on NIST 2008 SRE, 8conv1conv-10sec test setup.

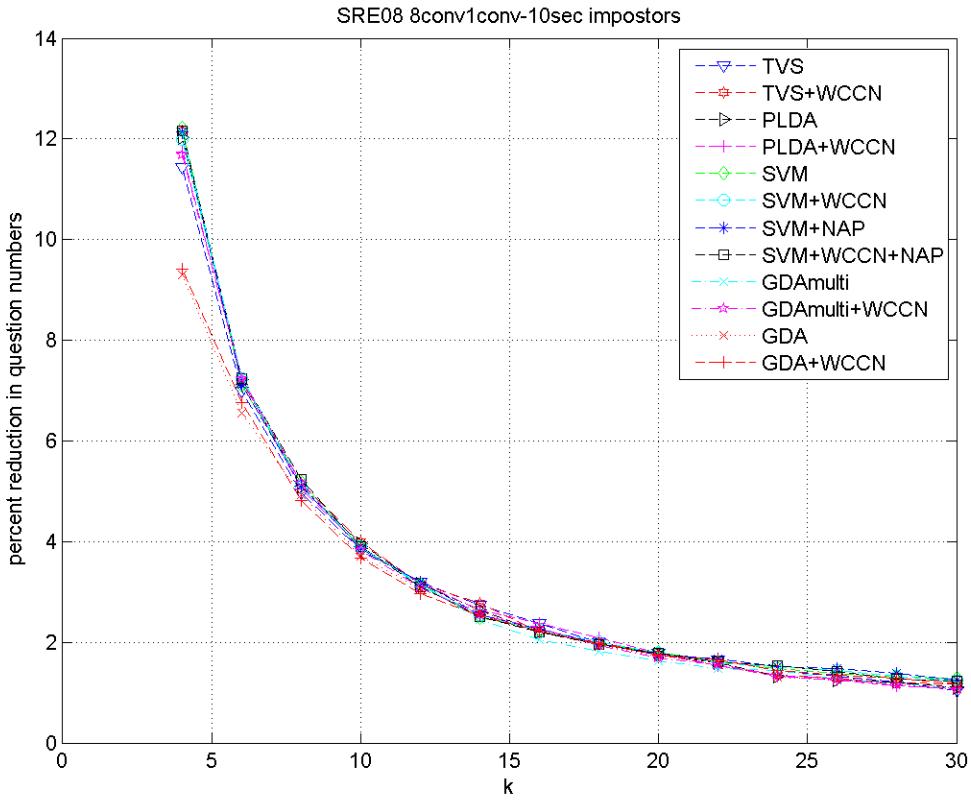

Figure 10: Reduction in number of questions with varying k values. Results are given on NIST 2008 SRE, 8conv1conv-10sec test setup.

CHAPTER IX

CONCLUSION

Semi-automatic speaker verification techniques have been proposed using some of the popular speaker verification algorithms. Firstly the performances of our individual systems on the NIST 2006 and 2008 database were compared. Then it was showed that, using the proposed methods, substantial reduction in the number of security questions needed by the human agent for verification can be obtained without an increase in the probability of false alarms. This result holds true especially for the clients which constitute the majority of the callers in a typical call center. Also, in par with the EER results, GDA system stands out as the best single option in most of the test cases. In conclusion, our results indicate that the current speaker verification systems can be effectively used in commercial applications that have tight security constraints if deployed in a semi-automatic fashion.

APPENDIX A

UTILIZED FUNCTIONS AND TOOLBOXES

- OzULibrary

  extractFeaturesOZU.m : A MATLAB function for feature extraction including different VADs

  UbmCreating_EM.m : A MATLAB wrapper function for UBM training

  MEX_ExpectationMaximization.cpp : A C++ class, implementation of EM algorithm

  estimateTparFastOZU.m : A MATLAB function for T training

  getIvectorOZU.m : A MATLAB function for i-vector extraction

  extractWCCNmodelUsingIvectorsOZU.m : A MATLAB function for WCCN matrix training

  PLDA_Train.m : A MATLAB function for PLDA model training

  getExpectedValuesPLDA.m : A MATLAB function for computing the expected values of latent variables

  PLDA_Verification.m : A MATLAB function for calculating the likelihood ratio given the enrollment and test data

  detCurve.m : A MATLAB wrapper function for DET-Curve plotting within EER and minDCF values via toolbox **DETware**

- Toolboxes

  libsvm : A well known open source library written in C for SVM training and test purposes

drtoolbox : An open source library written in MATLAB and C for GDA training and test purposes

SPro : A well known speech processing library written in C which provides feature extraction functions for speaker and speech recognition purposes

FoCal : A MATLAB library which provides fusion and calibration of automatic speaker detection systems via logistic regression methods

DETware : DET-Curve plotting software written in MATLAB

Bibliography

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, vol. 52, pp. 12–40, Jan 2010.

[2] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 430–451, Jan 2004.

[3] P. Kenny, "Joint factor analysis of speaker and session variability: theory and algorithms," tech. rep., CRIM, Jan 2006.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788–798, May 2011.

[5] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, Oct 2007.

[6] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, 2010.

[7] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," in *Proc. ˜Interspeech*, (Florence, Italy), Aug 2011.

[8] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, pp. 2385–2404, Oct 2000.

[9] V. Vapnik, *The Nature of Statistical Learning*. Springer, 1995.

[10] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. of ICSLP*, pp. 1471–1474, 2006.

[11] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. 97–100, May 2006.

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," in *Digital Signal Processing*, pp. 19–41, 2000.

[13] T. Hasan and J. Hansen, "A study on universal background model training in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 1890 –1899, Sept 2011.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing 2011 ed., Oct 2007.

[15] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, pp. 345 – 354, 2005.

[16] "Evaluation plan for nist sre 2004." `http://www.itl.nist.gov/iad/mig/tests/spk/2004/SRE-04_evalplan-v1a.pdf`. Accessed June 15, 2013.

[17] "Evaluation plan for nist sre 2012." `http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf`. Accessed June 15, 2013.

[18] `http://www.nuance.com/for-business/by-solution/customer-service-solutions/solutions-services/inbound-solutions/voice-authentication-biometrics/freespeech/index.htm`. Accessed June 15, 2013.

[19] N. Brummer, "Focal tool for logistic regression." `http://dsp.sun.ac.za/~nbrummer/focal/index.htm`. Accessed June 15, 2013.

[20] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[21] `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC97S62`. Accessed June 15, 2013.

[22] "Spro tool for feature extraction." `http://www.irisa.fr/metiss/guig/spro/spro-4.0.1/`. Accessed June 15, 2013.

VITA

Fatih Yeşil was born in Konya. After completing high school in Mersin, he started to study at İstanbul Technical University. He studied Telecommunications Engineering there.