JOINT SPARSE CODING OVER LEARNED DICTIONARIES AND THE USE OF LOW-COMPLEXITY PROJECTIONS FOR SPEAKER VERIFICATION

Haris B C

JOINT SPARSE CODING OVER LEARNED DICTIONARIES AND THE USE OF LOW-COMPLEXITY PROJECTIONS FOR SPEAKER VERIFICATION

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

Haris B C

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

AUGUST 2014

Certificate

This is to certify that the thesis entitled "**Joint Sparse Coding over Learned Dictionaries and the Use of Low-complexity Projections for Speaker Verification**", submitted by **Haris B C**, Roll No. 09610219, a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for the submission. The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

24th March 2015
Guwahati.

Dr. Rohit Sinha
Associate Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

To,

my father,

Sri. Ammad B C

and

my wife,

Lijiya

Acknowledgements

I express my deepest and most sincere gratitude to my thesis supervisor Dr. Rohit Sinha for his guidance and constant encouragement. His insightful feedbacks have helped me greatly in improving the quality of my thesis. I greatly admire his attitude towards research, creative thinking and enthusiasm for work.

I am grateful to Prof. S R M Prasanna, my doctoral committee chairman and the principal investigator of the project in which I have worked, for his support and encouragements throughout my PhD period.

I am thankful to the other members of my doctoral committee, Prof. S. Dandapat and Dr. Shaik Rafi Ahamed for their valuable suggestions on my work and for sparing their time for evaluating my work. I would like to thank Prof. P K Bora, Dr. A Rajesh and Dr. Tony Jacob and other faculty members of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their care and support.

I sincerely thank to Prof. B. Yegnanarayana, IIIT Hyderabad, for his valuable suggestions on my research work during the project review meetings, which helped me a lot in my research. I would like to acknowledge E-Security Division, Department of Information technology, New Delhi for funding my research.

I would like to express my sincere gratitude to Mr. Sanjib Das, Scientific officer for his enormous help whenever required. I am also grateful to all other technical and non-teaching staffs of the department and the computer center for their help. I am very much thankful to Dr. L. N. Sharma, Technical officer, and my seniors in EMST Lab Dr. Govind D, Dr. Sumitra Shukla, Dr. Debadatta Pati, Dr. S R Nirmala, Mr. Om P. Singh and Mr. Sunil Y for their help and support.

I am thankful to my friends Dr. Gayadhar Pradhan, Sayed Shahnawazuddin, Deepak K. T., Ramesh C. Mishra, K. K. Ramesh, Biswajit Dev Sarma, Nagaraj Adiga, Rohan Kumar Das, Ramesh K. Bhukya, Anurag Singh, Sibasankar Padhy, Malaya Kumar Nath, Sumit Shukla, Abhinav Misra and all other members in the EMST Laboratory.

My deepest gratitude goes to my parents and my wife. Without their love, support and sacrifice it wouldn't have been possible for me to complete my PhD.

Haris B C

Abstract

In the last few years, a number of speaker verification (SV) systems exploiting sparse representation classification (SRC) using different types of speaker representations have been proposed. The existing SRC based SV approaches use an exemplar dictionary created by arranging the training vectors of the target speaker and that of a set of background speakers as columns (atoms). The exemplar dictionary used for the SRC based approaches may not generalize well due to the absence of any learning unlike that used in the i-vector based approach. The SRC over exemplar dictionary based methods suffer from certain other shortcomings also, such as the need for selecting an optimal set of background speakers and the requirement of multiple examples for each speaker. Motivated by these we propose a novel SV paradigm based on the sparse coding of GMM supervectors over a learned dictionary. The derived sparse codes are used as speaker representations for SV task. This approach is further extended to enable a built-in compensation of session/channel variability with the use of joint sparse coding over learned speaker and channel dictionaries. The proposed sparse representation based SV system is compared with that of state-of-the-art i-vector PLDA system on the NIST 2012 SRE dataset and a relative performance improvement of 14.4% in terms of the detection cost function (DCF) is noted. In comparison to the i-vector based system, the salient attributes noted about the proposed SV system are: (i) a significantly enhanced performance for very low-false alarm rates, (ii) a higher robustness to the short duration test data condition, (iii) a competitive robustness to additive noise in test data, and (iv) a much lower computational complexity. With these features, the proposed approach seems to be a promising candidate for the practical high-security voice biometric applications.

Though the i-vector based approach is reported to result in very good performance, it is noted to have some drawbacks. The derivation of i-vectors is computationally complex and requires a large amount memory to store the transformation matrix and the algorithm-specific variables. These

problems could be addressed to some extent by obtaining the projections of GMM supervectors using low-complexity data-independent projections existing in literature. The low-complexity data-independent projection methods such as normal random projection, sparse random projection and simple decimation are explored in context of PLDA classifier based SV system. The resultant system is found to be attractive in terms of both the number of computations involved and the memory requirement, without much degradation in the performance compared to the default i-vector based one. A novel SV system that exploits the diversity among the representations obtained by using different offsets in the decimation of supervector, is also proposed. The proposed system is found to achieve 7% relative improvement in DCF over the i-vector based system on NIST 2012 SRE dataset while still having lesser overall complexity.

Keywords: Speaker verification, GMM mean supervectors, sparse representations, learned dictionary, discriminative learning, random projections.

Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial neural network
BKSVD	Block K-singular value decomposition
BP	Basis pursuit
CAT	Cluster adaptive training
C_{DET}	Detection cost
CDS	Cosine distance scoring
CMS	Cepstral mean subtraction
CSSOMP	Class supervised simultaneous orthogonal matching pursuit
CVN	Cepstral variance normalization
DLDA	Direct linear discriminant analysis
EER	Equal error rate
EM	Expectation maximization
FA	Factor analysis
GMM	Gaussian mixture model
GPLDA	Gaussian probabilistic linear discriminant analysis
HMM	Hidden Markov model
HPLDA	Heavy-tailed probabilistic linear discriminant analysis
HVAC	Heating ventilation and air conditioning
IID	Independent and identically distributed
i-vector	Identity vector
JDLD-SR	Sparse representation over joint discriminatively learned dictionary
JFA	Joint factor analysis
JL	Johnson-Lindenstrauss (lemma)
JLD-SR	Sparse representation over joint learned dictionary

KSVD	K-singular value decomposition
LARS	Least angle regression
LASSO	Least absolute shrinkage and selection operator
LC-KSVD	Label constraint K-singular value decomposition
LDA	Linear discriminant analysis
LD-SR	Learned dictionary (based) sparse representation
LLR	Log likelihood ratio
LPCC	Linear predictive cepstral coefficient
LSF	Line spectral frequency
MAP	Maximum a-posteriori
MFCC	Mel-frequency cepstral coefficient
min-C_{DET}	Minimum detection cost
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
MOD	Method of optimal directions
MODD-SV	Multi-offset decimation diversity speaker verification
MP	Matching pursuit
NAP	Nuisance attribute projection
NIST	National Institute of Standards and Technology (U.S. Dept. of Commerce)
NMF	Non-negative matrix factorization
OMP	Orthogonal matching pursuit
PCA	Principal component analysis
PDF	Probability density function
PLDA	Probabilistic linear discriminant analysis
PLP	Perceptual linear prediction
PSNR	Peak signal-to-noise ratio
RIP	Restricted isometric property
SAC	Sparse agglomerative clustering
SD	Standard deviation
SI	Speaker identification
S-KSVD	Supervised K-singular value decomposition

SNR	Signal-to-noise ratio
SRC	Sparse representation classification
SRE	Speaker recognition evaluation
SV	Speaker verification
SVM	Support vector machine
T-LD-SR	Total variability learned dictionary (based) sparse representation
T-matrix	Total variability matrix
T-norm	Test normalization
UBM	Universal background model
VAD	Voice activity detection
VQ	Vector quantization
WCCN	Within-class covariance normalization
XD-SR	Exemplar dictionary (based) sparse representation
Z-norm	Zero normalization

1

Introduction

Contents

Speaker recognition refers to the technology that enables machines to recognize persons using their voice samples. Based on the constraint imposed on the text content of the speech used, speaker recognition systems can be classified into text-dependent and text-independent ones. In text-dependent speaker recognition, each of the users speaks a pre-defined text which is known to the system. On the other hand, in text-independent speaker recognition no such constraint is used. Based on the mode of operation speaker recognition systems can be classified into *speaker identification* (SI) and *speaker verification* (SV) systems. In an SI system, the task is to identify the speaker of the input speech utterance from a set of enrolled speakers. In contrast, an SV system takes not only the speech samples but also an identity claim as inputs and decides whether the input speech belongs to the claimed speaker or not. The predominant applications of SI and SV technologies lie in person authentication and forensics. Among all the different variants of speaker recognition technologies mentioned above, text-independent SV is considered to be the most challenging one and is the focus of research in recent years. In last two decades, the SV technology has made a considerable progress and achieved an accuracy that makes it deployable for many practical applications.

1.1 Review of current SV approaches

Typical SV systems use short-term features for parameterization of speech and employ either a generative or a discriminative model for classification. The most successful short-term features are mel frequency cepstral coefficients (MFCC) [2] and perceptual linear predictive cepstral coefficients (PLPCC) [3]. For modeling the speakers, most commonly a Gaussian mixture model (GMM) [4] is used. In this approach, typically a few thousands of Gaussian components are used to capture the distribution of feature vectors of a given speaker and the parameters of the same are estimated using the available training data. For verifying a claim, the likelihood of the test data feature vectors are computed over the GMM that corresponds to the claimed speaker. Usually an approach involving the adaptation of the mean parameters of a universal background model (UBM) [5] using the speaker training data is followed to derive the speaker models. The UBM is a GMM trained using a sufficient amount of speech data from a large number of speakers by following the maximum likelihood (ML) approach. Thus it accounts for the speaker-independent distribution of speech feature vectors. For adaptation of the UBM mean parameters using the speaker dependent data, usually the maximum *a-posteriori* (MAP) approach is followed. This speaker modeling approach

is generally termed as GMM-UBM in short.

Support vector machine (SVM) is a discriminative classifier and is also widely used for speaker verification [6, 7]. The SVM is a binary classifier which uses a separating hyperplane as the decision boundary between the target speaker and the imposter (non-target speakers) population. The separating hyperplane that maximizes the margin of separation between these two classes is learned in the training phase. In the testing phase the decision is taken based on the distance of the test vector from the hyperplane that corresponds to the claimed speaker. For developing an SVM based SV system, an appropriate kernel function is used to map the speech utterances of arbitrary durations to a fixed and high-dimensional kernel space where the two classes are easier to separate with a hyperplane. In fact the linear hyperplane in the high-dimensional kernel space corresponds to a nonlinear decision boundary in the input feature vector space. There exists a number of kernels [1] which can be grouped into either parametric or derivative ones. Fisher kernel [8], GMM-supervector kernel [9], maximum likelihood linear regression (MLLR) kernel [10] and cluster adaptive training (CAT) based kernel [11] are a few to mention and among those the GMM-supervector kernel [12] happens to be the most commonly used one. A GMM supervector is derived by concatenating the mean parameters of all components of the speaker-adapted GMM-UBM model to form a high dimensional vector. Apart from individual kernel based approaches, the combination of multiple kernel functions has also been explored to improve the performance [13]. The SVM based SV systems have generally been found to outperform the traditional GMM-UBM likelihood based approaches.

Exploiting the fixed dimensional representation of the utterances provided by GMM supervectors, a number of session/channel variability compensation methods such as nuisance attribute projection (NAP) [14], linear discriminant analysis (LDA), within-class covariance normalization (WCCN) [15], joint factor analysis (JFA) [16] have also been developed. Majority of the current SV systems use a low dimensional representation of GMM supervectors called *i-vectors* [17], derived using factor analysis for representing the speaker utterances. The earlier i-vector based SV systems [17] used a simple cosine distance based classifier along with NAP/WCCN and LDA for channel compensation. Later in [18], a Bayesian approach referred to as probabilistic linear discriminant analysis (PLDA) that performs simultaneous channel compensation and classification was proposed. The PLDA based SV system uses a generative approach to model the speaker and channel subspaces in the i-vectors domain. The hyper-parameter matrices representing the

speaker and channel subspaces are learned offline using a labeled development data. Given an i-vector representing the training utterance of a target speaker, the latent variables corresponding to the speaker and channel subspaces are estimated. The likelihood of the test data i-vector is computed from the marginal density function obtained by integrating the channel factors out. In [18], heavy-tailed priors are assumed for the latent variables to handle the non-Gaussian nature of the i-vectors and the resultant model is known as heavy-tailed PLDA (HPLDA). In the simplified Gaussian PLDA (GPLDA) model proposed in [19], a radial Gaussianization (whitening) followed by length normalization is applied on the i-vectors and Gaussian priors for the latent variables are assumed. The GPLDA approach is reported to be much faster compared to HPLDA without any degradation in performance and is the most commonly pursued one at present.

In the last few years a lot of interest is generated about sparse representation of signals which provides new directions to signal processing research [20]. Sparse representation denotes the process of computing the sparsest solution among the infinitely many solutions available for a redundant system of linear equations. In sparse representation literature, the coefficient matrix of the linear system of equations is generally termed as *dictionary* and the columns of the dictionary are called as *atoms*. SR has been exploited successfully for various signal processing tasks like de-noising, compression etc. Recently the discriminative abilities of SR have also been exploited in various areas of applied pattern recognition. The signal classification using SR was first proposed by Huang *et. al.* [21] and later Wright *et. al.* [22] exploited it for the face recognition task. In the SR based classification, an exemplar dictionary is created by arranging the training examples corresponding to all classes in the task as atoms. A target vector is then approximated as a sparse linear combination of the atoms of the dictionary. The class assignment is done by comparing the absolute sums of the sparse coefficients corresponding to the atoms belonging to different classes. Motivated by the excellent performance reported for the face recognition task, the SR based classification is explored for the speaker identification task in [23] using GMM supervectors as speaker representations. Later in [24], the same approach is extended to the SV task using a set of background (non-target) speakers. For the verification of each claim, a dictionary is created using the examples of the claimed speaker along with that of the background speakers, and the test vector is sparse coded with that dictionary. In the sparse code, the coefficients corresponding to the claimed speaker atoms are compared to those corresponding to the imposter speaker atoms with a suitable metric [25]. In [26, 27], the same idea is explored with i-vectors as the speaker

representations. These SR based SV approaches are reportedly found to give competitive but lower performances in comparison to their corresponding baselines when evaluated on the NIST 2006 and 2008 speaker recognition evaluation (SRE) datasets. Unlike the above discussed approaches, in [28] an exemplar dictionary created directly using the MFCC features is used for the SR based SV task. The developed system is reported to give inferior but complimentary performance to the GMM-UBM based one and thus an improvement is obtained with the fusion of the two systems.

Apart form the above described approaches, there are a few works in literature exploiting non-negative matrix factorization (NMF) to obtain sparse representations for speaker recognition. In [29], non-negative tensor principal component analysis with a sparsity constraint in spectro-temporal domain is used to extract features for a closed-set speaker identification task. The proposed features showed improved noise robustness when compared to the traditional features like LPCC and MFCC. In [30], NMF of GMM supervectors is used to derive sparse and discriminative representations of speakers and the corresponding SV system is found to outperform a GMM-UBM baseline system on a Putonghua (Mandarin) corpus [31]. In [32], NMF is used to derive sparse vectors from the spectrograms of speech to represent speakers and these representations are then used to perform text-dependent speaker identification. The reported experiments show the higher noise robustness of the proposed system in comparison to the GMM-UBM based and the hidden Markov model (HMM) based baseline systems.

1.2 Motivation to the work

As mentioned in the above section, the current SV approaches predominantly use GMM mean supervectors as a fixed dimension representation of the speaker utterances. For reducing the dimensionality of the supervectors, the factor analysis based i-vector approach uses a projection matrix that represents the subspace where the principal variations in the data lie. In the i-vector framework proposed in [17], the learning of the subspace matrix as well as the computation of projection is done following maximum likelihood criterion. This approach has close similarity with principal component analysis (PCA) which is based on a minimum error criterion. In other words, the i-vectors can be interpreted as being derived minimizing the representation error. To enhance the discrimination between classes, generally the i-vectors are processed with various additional transformations such as NAP, LDA and WCCN. On the other hand, in the SR based approaches described above, a discriminative projection of the supervectors is intended. But, unlike the i-

vector based one, the SR based approaches use an exemplar matrix (dictionary) and that may not generalize well in absence of any learning. The existing SR base methods also suffer from certain other shortcomings such as the need for selecting an optimal set of background speakers and the requirement of multiple examples for each speaker. Interestingly, despite these issues the SR based approaches are reported to perform competitive to the i-vector based one.

Considering the above mentioned facts, we hypothesis that an SV system that can exploit the positive aspects of both the i-vector based and the SR based approaches would result in a better performance. The work reported in [33] tries to replicate the probabilistic JFA approach with a non-probabilistic signal coding framework. Though this work can be seen as the first step towards building the link between the factor analysis and sparse representation based approaches, it does not try to exploit the possible advantages in terms of discrimination achieved due to the sparsity in speaker representations. Motivated by these, in this work we first propose the use of a learned dictionary to address the shortcomings in the exemplar dictionary based approach and to exploit the generalization achieved with learning. The proposed approach is then extended by the use of a joint speaker-channel dictionary which provides a built-in compensation of the session/channel variability.

Despite the success of the i-vector approach in representing speaker utterances in low-dimensional space, it is noted to have some shortcomings too. The main problem lies in high complexity of the i-vector computation process which is further aggravated by requirement of a large amount of memory for the storage of transformation matrix and the algorithm-specific variables. In recent past, these issues have attracted the attention of researchers and a few works simplifying the i-vector computation are already reported in literature. An approach presented in [34] uses two approximations to reduce the number of computations. The first one is the use of a constant GMM component alignment across utterances given by the UBM-GMM weights. The second assumption is that the i-vector extractor matrix can be linearly transformed so that its per-Gaussian components are orthogonal. In [35] the factor analysis (FA) is performed on the pre-normalized centered GMM first-order statistics supervector to ensure that the statistics sub-vector of each of the Gaussian components is treated equally in the FA, which reduces the computational cost significantly. In addition the matrix inversion term is simplified using a look-up table based approach which resulted in further speed up with only a small quantization error. A fast i-vector computation using the factored sub-space approach is also proposed that achieves a five fold reduction in the

memory required for storing the T-matrix, without degradation in performance [36]. In literature data-independent projection approaches using random matrices are reported to provide a viable alternative to the data-dependent ones [37]. In this work we have also tried to reduce the computational burden in the front-end of SV system with the use of a few low-complexity data-independent projections.

1.3 Contributions in the thesis

The thesis mainly deals with the application of sparse representation techniques and exploration of some efficient data-independent projections for speaker verification. The salient contributions made in this thesis are summarized below:

- An SV system that uses sparse representation of GMM supervectors over a learned dictionary is proposed. Both simple and discriminative methods are explored for learning the dictionary in the supervector domain.
- For enabling built-in session/channel variability compensation, an approach using the joint sparse coding over speaker and channel dictionaries is developed.
- The proposed systems employing the SR over learned dictionary are evaluated on a large multi-variability dataset that includes test utterances recorded over multiple-channels, of varying durations and with ambient noise.
- For reducing the high computational complexity in the front-end of the present SV systems, the use of data-independent random and non-random (decimation) projections for the dimensionality reduction of GMM supervectors is explored.
- A novel SV system that exploits the diversity among the representations obtained by using different offsets in the decimation of the GMM supervectors is developed.

1.4 Organization of the thesis

The rest of the thesis is organized as follows. The Chapter 2 describes various modules of a typical SV system and reviews the most commonly used techniques for realizing each of the modules. The state-of-the-art SV approaches are also described in detail which are used to contrast the efficacy of the SR paradigm pursued in this work.

In Chapter 3, first the existing approaches of speaker identification and verification using SR over an exemplar dictionary in GMM supervector domain are described and major drawbacks

in them are highlighted. It is followed by the presentation of the proposed speaker verification approach exploiting the SR over learned dictionary. Dictionary learning using both simple and discriminative approaches are presented. For session/channel compensation in the proposed SR based systems, the supervectors are preprocessed with JFA prior to dictionary learning. In addition to the SR over exemplar dictionary based system, an i-vector based system with cosine kernel scoring and session/channel compensation with LDA and WCCN is also implemented and used as contrast system. The evaluation and analysis of the performance of the proposed as well as the contrast systems performed on the NIST 2003 SRE dataset, is also presented.

In Chapter 4, the proposed approach of SV using sparse representation over a learned dictionary has been extended to achieve built-in session/channel compensation. This is achieved with help of a novel approach developed exploiting the joint sparse coding over speaker and channel dictionaries. On contrasting with the i-vector PLDA system and the SR over exemplar dictionary and the learned dictionary based systems with session/channel compensation using JFA, the efficacy of the proposed joint sparse coding approach is highlighted. The various SV systems are built and evaluated on the NIST 2012 SRE dataset. The performance of the proposed system is analyzed for the utterance durations and noisy conditions in the test data. A comparison of the computational complexity of the proposed as well as the contrast systems are also provided.

Chapter 5 deals with the development of a low-complexity SV system by using data-independent projections for reducing the dimensionality of GMM mean supervectors. This is motivated by the high computational complexity and memory requirement involved in the i-vector representation based SV systems. Data-independent projections using random matrix, sparse random matrix and decimation operation are used. The concept is initially explored in context of a speaker identification system using SR based classification on the NIST 2005 SRE dataset. The low-complexity data-independent projections are then used in a PLDA based SV task. The experiments performed on the NIST 2012 SRE dataset to evaluate the performance of the proposed approaches are also presented. A novel SV system that exploits the diversity among the representations obtained by using different offsets in the decimation of supervector, is also proposed. The results are presented and the comparative analysis of the complexity of various systems are performed. Finally, the thesis is summarized and the future directions of the work are discussed in Chapter 6. The details of the key algorithms used for sparse coding and dictionary learning are presented in Appendix A and Appendix B, respectively.

2

Speaker Verification System: Structure and Techniques

Contents

The primary objective of this chapter is to provide the necessary context to the art of speaker verification by machines which forms the focus of the thesis. To serve the same, the various components of a typical text-independent SV system are described. The overall SV system comprises of two modules namely the training (enrollment) module and the testing (verification) module. The enrollment module of an SV system accepts the speech data from target speakers and creates the corresponding speaker models. The verification module accepts the test speech utterances and their corresponding speaker claims as inputs. It then verifies each of the claims and output its decisions by either accepting or rejecting it. Figure 2.1 shows the block-diagram of a generic SV system. The different blocks involved in the enrollment the verification modules are shown separately. The functionality of each of the blocks and the most commonly used techniques for realizing those are described in the following sections.

2.1 Signal processing and parameterization

The purpose of the signal processing and parameterization module is to remove silence portions from the raw speech and to transform the speech signal into feature vectors information about the speaker space is enhanced while suppressing the redundancies [38]. The task of removing silence from raw speech is usually performed with the help of a voice activity detector (VAD). Once the regions containing speech in an utterance are identified by the VAD, the remaining portions are treated as silence/non-speech and so are discarded. A simple VAD that uses signal energy to classify speech and silence regions works satisfactorily for telephone quality speech [1]. In that

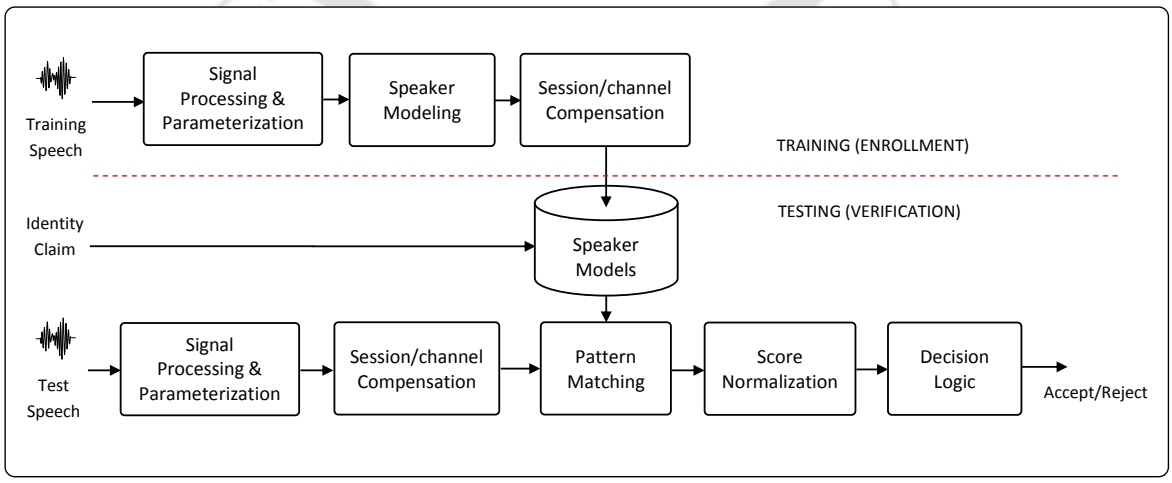


Figure 2.1: Block-diagram of a generic speaker verification system.

method, a threshold derived from the average energy of the utterance is compared with the energy of each of the signal frames to decide whether the corresponding frame belongs to speech or silence category. The major disadvantage of an energy based VAD is its sensitivity to additive noise. Among the many advancement proposed to address this problem, the simplest one is the use of two thresholds in the energy based VAD to enhance its effectiveness in noisy conditions [39]. The two thresholds, out of which one is used for noise and the other for speech, are kept adaptive to the noise level and are derived using a sliding mean calculation of noise signal energy. The noise signal energy $E_n(k)$ for the k^{th} frame of speech is computed as,

$$E_n(k) = \varphi_1 E_n(k-1) + (1-\varphi_1)E(k) \tag{2.1}$$

if the fame is identified as speech and as,

$$E_n(k) = \varphi_2 E_n(k-1) + (1-\varphi_2)E(k) \tag{2.2}$$

if the fame is identified as noise. φ_1 and φ_2 are the adaptation parameters and their typical values lies in the ranges of $[0.85, 0.95]$ and $[0.98, 0.999]$, respectively. The noise threshold $T_n(k)$ and speech threshold $T_s(k)$ for the k^{th} frame are defined as,

$$T_n(k) = E_n(k) + \eta_n \tag{2.3}$$

$$T_s(k) = E_n(k) + \eta_s \tag{2.4}$$

where η_n and η_s are the additive constants and their typical values fall in the ranges of $[0.1, 0.4]$ and $[0.5, 0.8]$, respectively. The first few frames in the utterance are assumed to be the noise frames and the thresholds are initialized accordingly. When the energy of a given frame is greater than the speech threshold, the frame is identified as speech and when the energy is lower than the noise threshold then it is identified as silence. The use of two thresholds form a hysteresis and avoids the problem of fast changes in the detection which are obtained if a single threshold is used. In addition to the short-time energy profile, the VADs based on spectral entropy [39] and negentropy [40] are also explored for very low SNRs. A recently proposed likelihood ratio based VAD [41] that trains speech and non-speech models on an utterance-by-utterance basis is reported to outperform the energy based VAD by a wide margin for noisy telephone and microphone speech.

Speech is a non-stationary signal and therefore it is processed in short frames of about 20-

30 milliseconds in duration. These speech frames are assumed to be stationary and appropriate acoustic parameter vectors are derived from each of the frames. The majority of the acoustic parameterization techniques used for SV task are based on spectral features. Such methods capture the short-term spectral envelope which is correlated to the timbre as well as the resonance properties of the supralaryngeal vocal tract. The commonly used short-term spectral features are MFCC [2], LPCC [42], line spectral frequency (LSF) [42], and perceptual linear prediction (PLP) coefficients [3], with MFFCs being the most popular among those. Before computing the features the speech frames are usually pre-emphasized to boost the higher frequencies and a applied with suitable window function in order to reduce spectral leakage effect due to the short-term processing. Hamming and Hanning windows are the most commonly used window functions. Recently the use of multi-taper window [43] and a windowing technique that inherently performs frequency domain differentiation [44] for short-term analysis are also shown to be advantageous for SV tasks.

For computing the MFCC features, the fast Fourier transform which is an efficient implementation of discrete Fourier transform is used to analyze of frequency content of each of the speech frames and the corresponding magnitude spectrum is extracted. The magnitude spectrum of a frame is then multiplied with a triangular filter-bank whose center frequencies are uniformly spaced on the Mel-scale to derive the nonlinearly (Mel) warped spectrum coefficients. A logarithmic compression of the magnitude of the Mel-warped spectrum followed by the inverse mapping with discrete cosine transform results the MFCC coefficients. The first and second order derivatives of the MFCC coefficients are often appended with the base features to incorporate some temporal information. The resultant feature vectors are attributed to represent the static and dynamic characteristics of the vocal tract and are reported to provide good verification accuracy [5]. To reduce the effect of convolutive acoustic channel from the feature vectors cepstral mean subtraction (CMS) and cepstral variance normalization (CVN) are also performed.

Voice source features those characterize the glottal excitation signal of voiced sounds carry speaker specific information and have been exploited for speaker verification [45–48]. The salient voice source features used for speaker verification include the fundamental frequency, the glottal pulse shape, the degree of vocal-fold opening and the duration of the vocal-fold closing phase. With an assumption that the glottal source and the vocal tract are independent of each other, the source signal can be estimated by first estimating the vocal tract filter parameters and then inverse filtering the original waveform. The estimation of the vocal tract filter parameters can effectively

be done using methods like linear prediction (LP).

Various parameterization techniques characterizing the prosodic features such as syllable stress, intonation patterns, speaking rate and rhythm are also been explored for speaker verification [49–51]. In addition to these, the high-level features such as a speakers characteristic vocabulary (idiolect) [52] are also used for speaker verification. The voice source feature, prosodic features and high level features generally produce lower speaker verification performance compared to the short-term spectral features, but these are noted to provide significant performance improvement on augmentation with the former [1].

2.2 Speaker modeling techniques

The feature vectors extracted from the training speech data of the target speakers are used to train the corresponding speaker models and those are stored in the database of the system. Based on the nature of the approach followed, speaker modeling techniques can be divided into parametric and non-parametric ones. In non-parametric approaches, a representation of the training examples of each of the target speakers is derived in the enrollment phase and used as the templates. During testing these target speaker templates are compared with the test examples to obtain a measure of similarity. Vector quantization (VQ) [53] and support vector machines (SVM) [6] are examples of nonparametric models used for text-independent SV. On the other hand, parametric approaches use stochastic models to characterize the generation of the speech feature vectors. In the enrollment phase, the parameters of target speaker models are estimated using their corresponding training data. In the verification phase, the likelihood of the test data over the claimed speaker model is computed and is used as the measure of similarity.

Another classification of speaker models based on the training paradigm used is into generative and discriminative categories. Generative models computes the posterior probability of the unknown speaker given the observed data by using a learned likelihood function and the Bayes rule. On the other hand, the discriminative modeling approaches tries to compute the posterior probability directly from the observed data by applying suitable discriminative rules. The speaker modeling approaches based on GMM and VQ fall in the generative category as these estimate the feature distribution for each speaker, while those based on artificial neural networks (ANNs) [54,55] and SVMs model the boundary between speakers so are examples of the discriminative category. A few of the most commonly used speaker modeling approaches are described in the following

subsections.

2.2.1 Gaussian mixture model

Gaussian mixture model (GMM) is composed of a finite number of multivariate Gaussian components and is used to model the probability density function (PDF) of the feature vectors. A GMM denoted by θ can be expressed as,

$$p(\boldsymbol{x}|\theta) = \sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \tag{2.5}$$

where, $\boldsymbol{x}$ is the d dimensional feature vector, C is the number of Gaussian components in the model and $\mathcal{N}(.)$ denotes the Gaussian density function. The parameters w_c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ represent the mixture weight, mean vector and covariance matrix of the c^{th} mixture component. Further, the multivariate Gaussian density function is defined as,

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_c|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)\right\}. \tag{2.6}$$

For using GMM as a speaker model, the parameters $\{w_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=1}^{C}$ are required to be estimated using the corresponding speaker's training data. These parameters are estimated following the maximum likelihood (ML) criterion with the help of expectation maximization (EM) algorithm. On account of the computational constraints and the limited amount of available training data per speaker, generally the diagonal covariance matrices are used in modeling [4].

2.2.2 Adapted universal background model GMM

Typical text-independent SV systems use GMM having few thousands of mixture components to achieve a reasonable performance. To ensure that all of the mixture components are well-trained, a large amount of speaker-specific data is needed. Finding that much data per speaker is not trivial and to overcome this challenge, a Bayesian adaptation based approach is proposed in [5] for training speaker models. In this approach, a speaker-independent GMM called universal background model (UBM) is used for deriving the speaker models. A UBM is trained using hundreds of hours of speech data obtained from a large number of speakers and it represents the distribution of speaker-space in general. To obtain a speaker-dependent GMM, the parameters of the UBM are adapted using the speaker-dependent training data by following the maximum *a-posteriori* (MAP) approach. Even though all the parameters can be adapted only the mean

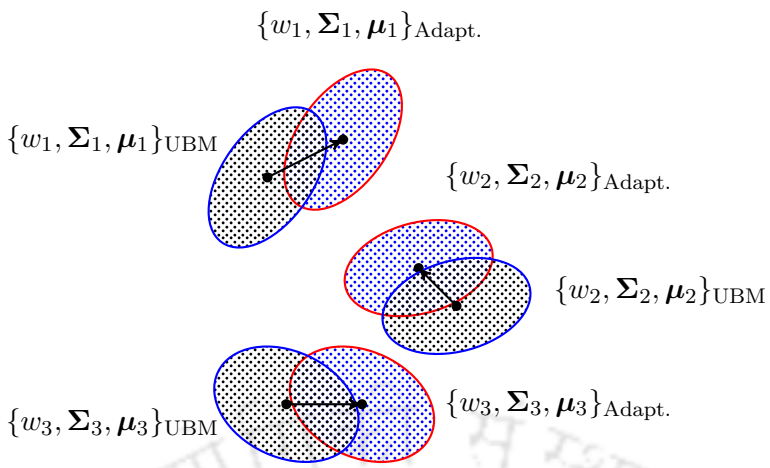


Figure 2.2: Illustration of the mean only adaptation of a UBM-GMM where the ellipses represent the multivariate Gaussians of a GMM having three components.

parameters are adapted in general as it is found to be effective.

Given a UBM parametrized by $\theta_{UBM} \equiv \{w_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=1}^{C}$, the MAP adaptation process of its mean parameters using the speaker-dependent data vectors $\boldsymbol{X} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ is performed as follows. The posterior probability $p(c|\mathbf{x_t}, \theta_{\mathbf{UBM}})$ of the mixture component c for a data vector $\mathbf{x_t}$ with respect to θ_{UBM} is computed as,

$$p(c|\boldsymbol{x}_t, \theta_{UBM}) = \frac{w_c \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{C} \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \tag{2.7}$$

Using $p(c|\boldsymbol{x}_t, \theta_{UBM})$ the sufficient statistics for estimating the mean are computed as,

$$0^{th} \text{ order statistics:} \quad N_c = \sum_{t=1}^{T} p(c|\boldsymbol{x}_t, \theta_{UBM}) \tag{2.8}$$

$$1^{st} \text{ order statistics:} \quad \boldsymbol{F}_c = \frac{1}{N_c} \sum_{t=1}^{T} p(c|\boldsymbol{x}_t, \theta_{UBM}) \boldsymbol{x}_t \tag{2.9}$$

Then, the MAP adapted mean vector $\boldsymbol{\mu}'_c$ corresponding to mixture c is computed as,

$$\boldsymbol{\mu}'_c = \alpha_c \boldsymbol{F}_c + (1 - \alpha_c)\boldsymbol{\mu}_c \tag{2.10}$$

where α_c is the parameter that controls the contributions of the *prior* and the data. Fig. 2.2 shows an illustration of mean only adaptation of a 3-component GMM of two-dimensional data.

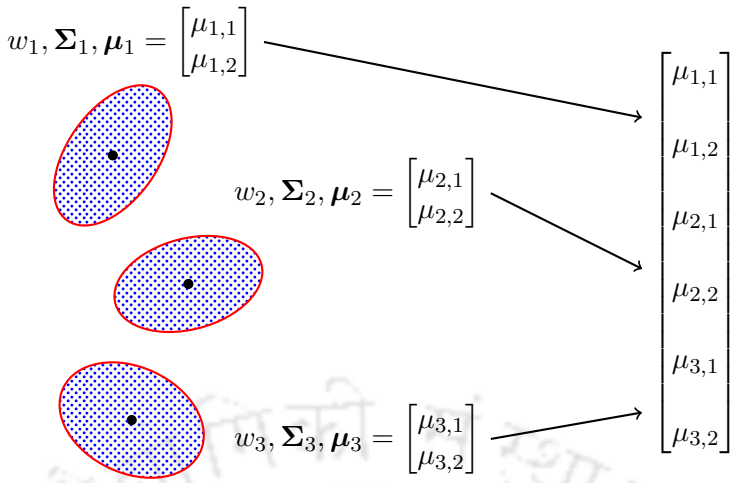


Figure 2.3: Illustration of extraction of a GMM mean supervector where the ellipses represent the multivariate Gaussian of an adapted GMM. In this work only adapted mean parameters of the Gaussian components are used for deriving the supervectors and no normalization by the $\sqrt{w_i}\,\Sigma_i^{-1/2}$ is incorporated as suggested in [1].

2.2.3 GMM mean supervector representation

Apart from the likelihood based approaches, the SVM classifier is also been widely explored for the SV task [6, 12, 56]. For developing an SVM based SV system, the speech utterances of arbitrary durations is needed to be mapped to a fixed and high-dimensional space. This is generally performed with the help of an appropriate kernel function. There exists a number of parametric and generative kernel functions for this task. Fisher kernel [8], GMM-supervector kernel [9], maximum likelihood linear regression (MLLR) kernel [10] and cluster adaptive training (CAT) based kernel [11] are a few to mention and among those the GMM-supervector kernel [12] happens to be the most commonly used one. GMM mean supervectors are created by concatenating the mean vectors corresponding to a speaker adapted GMM-UBM and it provides an effective way of representing a speaker utterance of arbitrary duration in terms of a fixed dimension vector. Figure 2.3 illustrates the creation of a GMM mean supervector from an adapted GMM. Such a representation has opened the possibility of the use of many other classifiers other than SVM such as simple cosine distance and Bayesian classifier with probabilistic linear discriminant analysis (PLDA) modeling for speaker verification [18,19]. The advent of GMM mean supervector also led to the development of many highly effective session/channel compensation techniques [15, 16, 57] which are discussed later in this chapter.

2.2.4 i-vector representation

The GMM mean supervectors are found to be effective in representing speaker utterances, but are noted to be highly redundant in terms of speaker-dependent information. To reduce redundancy in the representation, commonly a factor analysis (FA) based approach is used in the front-end of SV systems [17]. In factor analysis, a given supervector is decomposed into an utterance-independent component and a low-rank utterance-dependent component. Thus a GMM mean supervector $\boldsymbol{y}$ is modeled as,

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{T}\boldsymbol{w} + \boldsymbol{\epsilon} \tag{2.11}$$

where $\boldsymbol{\mu}$ is the utterance-independent component, $\boldsymbol{T}\boldsymbol{w}$ is the utterance-dependent component and ϵ is the modeling error. Generally the UBM mean supervector is substituted for μ. $\boldsymbol{T}$ is the low-rank *total-variability matrix* (T-matrix) that represents the dominant speaker and channel variabilities present in the supervector. $\boldsymbol{w}$ is the *factors* or the *latent variable* vector and is assumed to have a standard Gaussian prior distribution.

The $\boldsymbol{T}$ matrix is learned iteratively using a large amount of speech data with the help of the EM algorithm. Given $\boldsymbol{T}$ and the UBM, the MAP point estimate of the latent variable $\boldsymbol{w}$ corresponding to a supervector $\boldsymbol{y}$ is computed as the mean of the corresponding posterior distribution and the closed form solution for the same is given as,

$$\hat{\boldsymbol{w}} = (\boldsymbol{I} + \boldsymbol{T}'\boldsymbol{\Sigma}^{-1}\boldsymbol{N}\boldsymbol{T})^{-1}\boldsymbol{T}'\boldsymbol{\Sigma}^{-1}\boldsymbol{N}\tilde{\boldsymbol{F}} \tag{2.12}$$

where, $\boldsymbol{N}$ and $\boldsymbol{\Sigma}$ are matrices whose diagonal blocks are $N_c\boldsymbol{I}$ and $\boldsymbol{\Sigma_c}$ respectively, and $\boldsymbol{I}$ is the identity matrix. N_c is the 0^{th} order statistics derived from data for on the c^{th} component of the UBM and $\boldsymbol{\Sigma}_c$ is the covariance matrix of the c^{th} component of the UBM. The vector $\tilde{\boldsymbol{F}}$ is obtained by concatenating the component-specific centered 1^{st} order statistics which is computed as,

$$\tilde{\boldsymbol{F}}_c = \frac{1}{N_c}\sum_{t=1}^{T} p(c|\boldsymbol{x}_t, \theta_{UBM})(\boldsymbol{x}_t - \boldsymbol{\mu}_c). \tag{2.13}$$

The variables in Equation 2.13 are same as that defined in Subsection 2.2.2. The estimated latent variable $\tilde{\boldsymbol{w}}$ is referred to as the *identity vector* (i-vector) and it is treated as a compact and fixed dimensional representation of the speaker utterance.

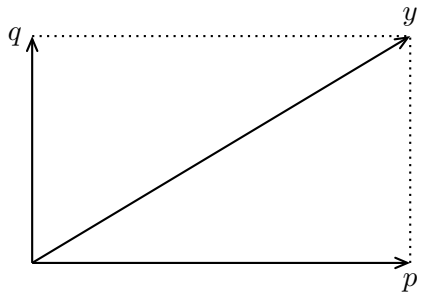


Figure 2.4: The joint factor analysis of speaker and channel variability is based on a decomposition of the form $\boldsymbol{y} = \boldsymbol{p} + \boldsymbol{q}$. The vector $\boldsymbol{y}$ is the GMM mean supervector representation for a given recording, $\boldsymbol{p}$ is the speaker component and $\boldsymbol{q}$ is the channel component.

2.3 Session/channel compensation methods

The performance of SV systems usually get severely affected due to the mismatch of session, recoding channel and environment conditions between the training and test data. These mismatches are generally termed as the *session/channel variability* in literature. The compensation for these have become an integral part of the current SV systems to achieve good performance in practical conditions. In the following the salient session/channel variability compensation methods that are commonly used are described briefly.

2.3.1 Joint factor analysis

Joint factor analysis (JFA) uses a generative model to separate the speaker and session/channel components in the GMM supervector representation of a speaker utterance [16]. It is assumed that the speaker and channel spaces are orthogonal to each other and hence a supervector $\boldsymbol{y}$ can be decomposed into a the speaker-dependent and a channel-dependent components as,

$$\boldsymbol{y} = \boldsymbol{p} + \boldsymbol{q} \tag{2.14}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the speaker and channel components of the supervector, respectively. The channel subspace is assumed to be low-rank and hence the component $\boldsymbol{q}$ is explicitly modeled as,

$$\boldsymbol{q} = \boldsymbol{U}\boldsymbol{u} \tag{2.15}$$

where $\boldsymbol{U}$ is a rectangular matrix representing the the channel subspace and $\boldsymbol{u}$ is the channel factor vector corresponding to the given utterance. Further, the speaker component $\boldsymbol{p}$ is assumed to have a speaker and channel independent component $\boldsymbol{\mu}$, a low-rank (eigen-voice) component $\boldsymbol{V}\boldsymbol{v}$ and a

full-rank component $\boldsymbol{Dd}$ as,

$$\boldsymbol{p} = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{v} + \boldsymbol{D}\boldsymbol{d} \tag{2.16}$$

where $\boldsymbol{V}$ is a low-rank matrix, and $\boldsymbol{D}$ is a diagonal matrix, both representing the speaker variability. The vectors $\boldsymbol{v}$ and $\boldsymbol{d}$ represents the corresponding factors. The component $\boldsymbol{Dd}$ is found to be less effective in limited data conditions and is avoided in general. The resulted simplified JFA model becomes,

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{u} + \boldsymbol{V}\boldsymbol{v} \tag{2.17}$$

The UBM supervector is a good approximation for the speaker independent component $\boldsymbol{\mu}$. In the parameter estimation stage, a maximum likelihood based discriminative training is used to learn the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ using large amount of labeled speaker data. The matrices are learned in an iterative manner using EM algorithm [58]. Given the parameters of the JFA model, the speaker and channel factors can be jointly estimated by using the matrix obtained by augmenting $\boldsymbol{U}$ and $\boldsymbol{V}$ instead of the $\boldsymbol{T}$ matrix in Equation 2.12. The channel compensated supervector is given by $\boldsymbol{V}\hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{v}}$ is the estimate of the speaker factors.

2.3.2 Linear discriminant analysis

Linear discriminant analysis (LDA) is used to perform session/channel compensation of i-vector representations by enhancing the separability between speakers. The data vectors are projected down to a set of new orthogonal axes where the discrimination between different classes (speakers) is maximum. The LDA projection matrix is composed by the eigen vectors corresponding to the most significant eigen values of the eigen analysis equation,

$$(\boldsymbol{W}^{-1}\boldsymbol{B})\boldsymbol{z} = \lambda\boldsymbol{z} \tag{2.18}$$

where $\boldsymbol{W}$ is the within-class covariance matrix, $\boldsymbol{B}$ is the between-class covariance matrix, $\boldsymbol{z}$ is the variable representing eigen vector, and λ is the corresponding eigen value [17].

2.3.3 Within-class covariance normalization

Within-class covariance normalization (WCCN) is another linear transformation method widely used to compensate the effects of session/channel mismatch in the supervector and the i-vector representations [15,17]. The transformation minimizes the upper bounds on the classification error metric and hence minimizes the classification error. The transformation matrix $\boldsymbol{B}$ is obtained by

Cholesky decomposition of the inverse of the within-class covariance matrix $\boldsymbol{W}$ as,

$$\boldsymbol{W}^{-1} = \boldsymbol{B}\boldsymbol{B}^{T} \tag{2.19}$$

2.3.4 Nuisance attribute projection

In nuisance attribute projection (NAP) method objective is to find the nuisance subspace which represents the channel/session variabilities [56]. The data is projected to the complimentary space of the nuisance subspace for compensation. The projection matrix P is given as,

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{T} \tag{2.20}$$

where R is a rectangular matrix containing the eigen vectors corresponding to few of the top eigen values of the within-class covariance matrix.

2.4 Pattern matching and score normalization

The pattern matching module choses the speaker model according to the identity claim input and compute the verification score of the test data on the selected model. This verification score is the measure of confidence of the system in the hypothesis that the input speech utterance is generated by the claimed speaker. The method of computing the verification score depends on the kind of the speaker modeling technique used. The score computation methods that are commonly used with various speaker modeling approaches, discussed in Section 2.2, are described in the following.

2.4.1 Likelihood ratio scoring with GMM modeling

In SV systems using parametric speaker modeling approaches like GMM or GMM-UBM, the likelihood of the test data vectors on the claimed speakers model is used as the score for verification. Usually, the feature vectors of a test utterance $\boldsymbol{X}$ are assumed to be independent and identically distributed (IID), so the likelihood of a claimed speaker model θ_c for a sequence of feature vectors, $\boldsymbol{X} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, is computed as,

$$p(\boldsymbol{X}|\theta_c) = \prod_{n=1}^{N} p(\boldsymbol{x_n}|\theta_c) \tag{2.21}$$

The likelihoods over the claimed model is often normalized with the likelihood on the UBM model to obtain the log-likelihood ratio (LLR) score as,

$$\begin{aligned} \text{LLR Score} &= \log \frac{p(\boldsymbol{X}|\theta_c)}{p(\boldsymbol{X}|\theta_{UBM})} \\ &= \log \frac{\prod_{n=1}^{N} p(\boldsymbol{x_n}|\theta_c)}{\prod_{n=1}^{N} p(\boldsymbol{x_n}|\theta_{UBM})} \\ &= \sum_{n=1}^{N} \log p(\boldsymbol{x_n}|\theta_c) - \sum_{n=1}^{N} \log p(\boldsymbol{x_n}|\theta_{UBM}) \end{aligned} \tag{2.22}$$

2.4.2 Cosine distance scoring

With GMM mean supervectors or i-vectors as the speaker utterance representations, SVM can be used for pattern matching [17, 56]. The representations for both the training and test speech utterance are derived and the SVM classifier is used to compute the verification score. Later it was shown in [59] that the use of a simple cosine distance measure between training and test data representations provides better performance compared to the SVM classifier for SV tasks. This is due to the fact that, in most of the publicly available SV databases like NIST SRE datasets, only a few utterances are is available to train each speaker. As a result of this, the training of SVM becomes suboptimal and it fails in achieving its effectiveness in such conditions. The cosine distance score is computed as,

$$\text{Score} = \frac{< \hat{\boldsymbol{w}}_{clm}. \hat{\boldsymbol{w}}_{tst} >}{\|\hat{\boldsymbol{w}}_{clm}\|_2 \ \|\hat{\boldsymbol{w}}_{tst}\|_2} \tag{2.23}$$

where $\hat{\boldsymbol{w}}_{clm}$ and $\hat{\boldsymbol{w}}_{tst}$ represent the representations of the training and the test utterances, respectively, and the operators have the usual meaning.

2.4.3 Bayesian classifier with probabilistic linear discriminant analysis

Probabilistic linear discriminant analysis (PLDA) [60] is a stochastic approach which is recently been explored for performing SV task in the i-vector domain [18, 19]. In PLDA, an i-vector $\boldsymbol{w}$ corresponding to a speaker utterances is represented using a generative model as,

$$\boldsymbol{w} = \boldsymbol{\rho} + \boldsymbol{H}\boldsymbol{h} + \boldsymbol{G}\boldsymbol{g} + \boldsymbol{\varepsilon} \tag{2.24}$$

where, $\boldsymbol{\rho}$ is the global mean of i-vector population, $\boldsymbol{H}$ and $\boldsymbol{G}$ are the matrices representing the speaker and channel subspaces respectively, $\boldsymbol{h}$ and $\boldsymbol{g}$ are the speaker and channel factors, respectively with each having standard normal prior distribution, and $\boldsymbol{\epsilon}$ is the residual factor having

standard normal prior with diagonal covariance. In [18], heavy-tailed priors are assumed for the latent variables for handling the effect of outliers in the data and the model is known as heavy-tailed PLDA (HPLDA). Later in [19] a simplified version of PLDA has been proposed which ignores the term $\boldsymbol{Gg}$ and assigns standard normal prior to the latent variables. The residual term is modeled with Guassian distribution with zero mean and non-diagonal covariance denoted by $\boldsymbol{S}$. A Gaussianization process consisting of whitening followed by length normalization is performed on the i-vectors prior to modeling and testing. This approach is commonly referred to as Gaussian PLDA (GPLDA) and reported to provide performance similar to that of the HPLDA with a much lesser complexity.

The parameters of the GPLDA model $\{\boldsymbol{\rho}, \boldsymbol{H}, \boldsymbol{S}\}$ are learned with the help of EM algorithm on a development dataset. Given the length normalized i-vectors $\boldsymbol{w}_{clm}$ and $\boldsymbol{w}_{tst}$ corresponding to the claimed speakers training utterance and the test utterance, respectively, the log-likelihood ratio score using the GPLDA model is computed by a hypothesis test as,

$$\begin{aligned}
\text{LLR Score} &= \log \frac{p(\boldsymbol{w}_{clm}, \boldsymbol{w}_{tst}|\Gamma_s)}{p(\boldsymbol{w}_{clm}|\Gamma_d)p(\boldsymbol{w}_{tst}|\Gamma_d)} \\
&= \frac{\int p(\boldsymbol{w}_{clm}, \boldsymbol{w}_{tst}, \boldsymbol{h}_{clm}|\rho, \boldsymbol{S})d\boldsymbol{h}_{clm}}{\int p(\boldsymbol{w}_{clm}\boldsymbol{h}_{clm}|\rho, \boldsymbol{S})d\boldsymbol{h}_{clm} \int p(\boldsymbol{w}_{tst}, \boldsymbol{h}_{tst}|\rho, \boldsymbol{S})d\boldsymbol{h}_{tst}} \\
&= \frac{\int p(\boldsymbol{w}_{clm}, \boldsymbol{w}_{tst})|\rho, \boldsymbol{S})p(\boldsymbol{h}_{clm})d\boldsymbol{h}_{clm}}{\int p(\boldsymbol{w}_{clm}|\rho, S)p(\boldsymbol{h}_{clm})d\boldsymbol{h}_{clm} \int p(\boldsymbol{w}_{tst}|\rho, \boldsymbol{S})p(\boldsymbol{h}_{tst})d\boldsymbol{h}_{tst}} \\
&= \frac{\mathcal{N}\left(\left[\boldsymbol{w}_{clm}^T \boldsymbol{w}_{tst}^T\right]^T | \left[\boldsymbol{\rho}^T \boldsymbol{\rho}^T\right]^T, \tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^T + \tilde{\boldsymbol{S}}\right)}{\mathcal{N}\left(\left[\boldsymbol{w}_{clm}^T \boldsymbol{w}_{tst}^T\right]^T | \left[\boldsymbol{\rho}^T \boldsymbol{\rho}^T\right]^T, diag\left\{\boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{S}, \boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{S}\right\}\right)}
\end{aligned} \tag{2.25}$$

where Γ_s is the hypothesis that both $\boldsymbol{w}_{clm}$ and $\boldsymbol{w}_{tst}$ share the same speaker identity latent variable $\boldsymbol{h}$ while Γ_d is the hypothesis that $\boldsymbol{w}_{clm}$ and $\boldsymbol{w}_{tst}$ are generated by different speaker identity latent variables $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$. $\tilde{\boldsymbol{H}} = \left[\boldsymbol{H}^T \boldsymbol{H}^T\right]^T$ and $\tilde{\boldsymbol{S}} = diag\left\{\boldsymbol{S}, \boldsymbol{S}\right\}$, where the operator $diag\{.,.\}$ creates a block diagonal matrix by placing the arguments across the diagonal. The final closed-form solution for the LLR score is derived as,

$$\text{LLR Score} = C + \boldsymbol{w}_{clm}^T \boldsymbol{Q} \boldsymbol{w}_{clm} + \boldsymbol{w}_{tst}^T \boldsymbol{Q} \boldsymbol{w}_{tst} + 2\boldsymbol{w}_{clm}^T \boldsymbol{P} \boldsymbol{w}_{tst} \tag{2.26}$$

where

$$C = \frac{1}{2}\ln\|\boldsymbol{D}_0\| - \frac{1}{2}\ln\|\boldsymbol{D}_1\| + \boldsymbol{\rho}^T(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{\rho}$$

$$\boldsymbol{P} = \Delta^{-1}\Phi(\Delta - \Phi\Delta^{-1}\Phi)^{-1}$$

$$\boldsymbol{Q} = \Delta^{-1} - (\Delta - \Phi\Delta^{-1}\Phi)^{-1}$$

$$\Delta = \boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{S}$$

$$\Phi = \boldsymbol{H}\boldsymbol{H}^T$$

$$\boldsymbol{D}_0 = \begin{bmatrix} \Delta & \Phi \\ \Phi & \Delta \end{bmatrix}, \ \boldsymbol{D}_1 = \begin{bmatrix} \Delta & \boldsymbol{0} \\ \boldsymbol{0} & \Delta \end{bmatrix}$$

2.5 Score normalization and calibration

The main purpose of score normalization is to transform scores from different speakers into a similar range so that a common (speaker-independent) verification threshold can be used. Score normalization can correct some speaker-dependent score offsets not compensated by the feature and/or the model domain methods like CMN, CVN and JFA etc. The normalizes score t' for a verification score t is computed as follows.

$$t' = \frac{t-m}{v} \tag{2.27}$$

where m and v are the mean and variance computed from a set of imposter scores, respectively.

The zero normalization (Z-norm) and test normalization (T-norm) are the two popular score normalization approaches. In Z-norm, the impostor statistics m and v are are target speaker dependent and they are computed off-line in the speaker enrollment phase. This is done by scoring a set of non-target (imposter) speaker utterances against the target model and obtaining the mean and standard deviation of those scores. In T-norm, the parameters are test utterance dependent and hence they are estimated at the time of verification. This is done by matching the unknown speakers feature vectors against a set of impostor models and obtaining the statistics [1]. The verification scores are often calibrated for better comparison against a fixed threshold or for the purpose of fusing with scores generated by another classifier. Score calibration is particularly important for the systems those generate a non-likelihood scores (e.g. CDS score) whereas the likelihood scores are required for decision making. In such cases the calibration process is used to

map the obtained scores to the likelihood values and the process involves a scaling and shifting operation. The parameters for calibration can be learned effectively form a set of *development trial* scores using the logistic regression. BOSARIS [61] is a well-known toolkit used for performing the logistic regression based score calibration.

2.6 Summary

In this chapter, the structure of a typical speaker verification system is described. The state-of-the-art speaker representation approaches viz. GMM mean supervector, factor analysis based i-vector and the corresponding classifiers used are explained in detail. Various session/channel compensation and score normalization techniques are also presented. Some of the discussed approaches and techniques are used in later chapters to provide the contrast to the different sparse representation based SV approaches explored in this thesis.

3

Sparse Representation over Learned Dictionary based Speaker Verification

Contents

In the last few years, a number of speaker recognition systems exploiting sparse representation classification (SRC) using different types of speaker representations have been proposed. These works are motivated by the success of SRC for the face recognition task [22] especially in noisy and occluded conditions. The earliest work [23] explored the sparse representation over an exemplar dictionary for speaker recognition. In that work, the exemplar dictionary is created by arranging the GMM mean supervectors corresponding to the speaker utterances in the training dataset as columns. The supervector representing the test utterance is represented as the sparse linear combination of the columns of that dictionary. The test supervector is assigned to the class associated to the atoms having the highest non zero coefficients in the sparse vector. Later the SRC with exemplar dictionary based approach is extended to the SV task in [24]. In that work, the dictionary for verifying a claim is constructed by arranging the GMM mean supervectors of the claimed speaker utterances and those of a set of imposter (non-target) speaker utterances. So constructed dictionary is then used for representation of the supervector corresponding to test utterance as a sparse linear combination of its atoms. For verification purpose, the coefficients of the sparse vector corresponding to the target speaker vectors is compared to those of the imposter speaker vectors with a suitable metric. The similar idea was explored with exemplar dictionary created using the total variability i-vectors in [26, 27] and using the MFCC feature vectors in [28]. Among the various vector representations of speech explored for SRC based speaker verification as of now, the GMM mean supervector has been found to be the most successful one.

On the other hand, state-of-the-art SV systems predominantly use the i-vector framework that is based on the modeling of the total variability subspace in which the significant variations of the GMM supervectors lie [17]. Similar to the PCA based representation, in the i-vector representations of the speaker utterances also happen to minimize the representation error only. As a result these are often further processed with additional transformations (e.g., LDA and WCCN) to enhance the discrimination. In contrast to that the SRC based approaches described above intends to find a discriminative projection of the supervectors. But unlike the i-vector based approach, the existing SRC based approaches employ a hand-crafted exemplar dictionary and hence those may not generalize well due to no learning being involved in the dictionary creation process. The existing SRC over exemplar dictionary based methods also suffer from certain other shortcomings such as the need for selecting an optimal set of background speakers and the requirement of multiple examples for each of the speakers. Interestingly, despite these issues the SR over exemplar based

SV approaches are reported to perform competitive to the i-vector based one.

Motivated by the above mentioned facts, we propose the use of a learned speaker dictionary to address the shortcomings in the exemplar dictionary based SV approach. For learning the speaker dictionary, the well-known K-means singular value decomposition (KSVD) [62] and a discriminative variant of KSVD are employed. The SRC over exemplar dictionary based and the simple CDS based SV systems, using both supervectors and i-vectors as representations, are used to contrast the performance of the proposed approach. The performance of the proposed system and that of the contrast systems evaluated and compared on the NIST 2003 SRE dataset.

The rest of this chapter is organized as follows. Section 3.1 provides the basics of sparse representation of signals. The existing works on the sparse representation over exemplar dictionary based SI and SV are reviewed in Section 3.2. The proposed learned dictionary based SV approach is presented in Section 3.3. The details of the experimental setup are given in Section 3.4 and the results are presented in Section 3.5. The chapter is concluded in Section 3.7.

3.1 Sparse representation of signals

Sparse representation refers to the representation of a target signal/vector as the sparse linear combination of the prototype signals (atoms) in a redundant dictionary. It can be achieved by the computation of the solution that contains the least number of non-zero elements to the corresponding redundant linear system of equations. The sparse coding of a target signal over a dictionary is illustrated in Figure 3.1. Consider that a dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ that contains n atoms, $\{d_j\}_{j=1}^{n}$, is being used to sparse code a signal vector $\boldsymbol{y}$. The solution (sparse code) $\boldsymbol{s}$ to the corresponding linear system of equations can be either exact, satisfying $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{s}$ or approximate satisfying $\boldsymbol{y} \approx \boldsymbol{D}\boldsymbol{s}$, satisfying $\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{s}\|_2 \leq \epsilon$, where ϵ is the error bound [62].

In case of an overcomplete dictionary, i.e., the dictionary having more columns than rows ($m < n$), and if $\boldsymbol{D}$ is full rank, there exist infinite number of solutions to the representation problem and to choose the sparsest one among them, the l_0-norm constraint is used. Thus the sparse representation problem can be formulated either as,

$$\arg\min_{\boldsymbol{s}} \|\boldsymbol{s}\|_0 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{D}\boldsymbol{s} \tag{3.1}$$

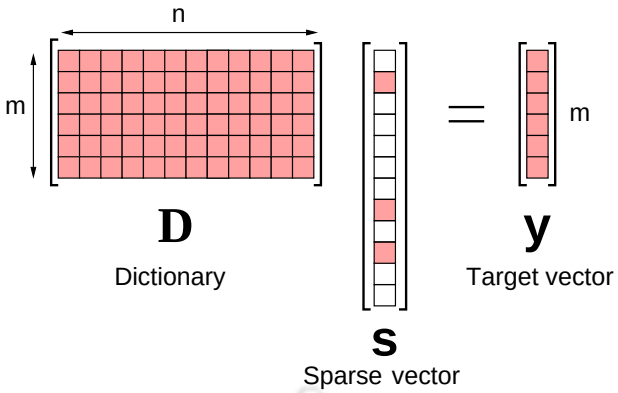


Figure 3.1: Illustration showing sparse coding of a target vector over a dictionary.

or as,

$$\arg\min_{\boldsymbol{s}} \|\boldsymbol{s}\|_0 \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{s}\|_2 \leq \epsilon \tag{3.2}$$

The signal processing applications in which the sparse representation is successfully employed include image de-noising [63], image compression [64], image and audio impainting [65, 66] and signal classification [21]. For developing the sparse representation framework for a type of signal, the two fundamental issues that are required to be addressed are: (i) the choice/creation of a dictionary suitable for the task, and (ii) the sparse coding problem, i.e., the computation of the sparse solution given the dictionary and the target signal.

Obtaining an exact solution to the sparse coding problem as given in Equation 3.1 is proven to be an NP hard. Thus the approximate solution is sought and there exist a number of pursuit algorithms for that purpose. Among those, the matching pursuit (MP) [67] and orthogonal matching pursuit (OMP) [68] algorithms are very simple yet quite effective. These algorithms are greedy in nature as the most suitable dictionary atoms are selected sequentially. The atoms are selected based on the value of the inner product between the target vector and the dictionary atoms, and the least square solution with the selected atoms provides the non-zero coefficients in the sparse solution. The MP and OMP algorithms differ in the fact that in the former an atom can be selected multiple times while the same is avoided in the former due to explicit orthogonalization. Basis pursuit (BP) [69] is another well-known pursuit algorithm used for computing sparse representations. It replaces the l_0 norm constraint in the sparse coding problem by l_1 norm to make the optimization task convex. An equivalent approach called least absolute shrinkage and selection operator (LASSO) uses a Lagrange multiplier to convert the constraint to a penalty term

and solves the problem using the least angle regression (LARS) algorithm [70].

The dictionaries used for sparse representation can, in general, be either parametric or data-driven. The parametric dictionaries, also known as analytic dictionaries, use pre-specified set of mathematical functions to represent the target data. The examples of analytic dictionaries include Fourier [71], wavelet [72], curvelet [73] and contourlet [74] transform based dictionaries. In many cases such dictionaries lead to simple and fast algorithms which do not involve multiplication by the dictionary matrix for the evaluation of the sparse representation [71]. The major drawback of the analytic dictionaries lies in the limitations of the model functions used in them. These models are typically too simplistic to handle the complexity of many natural signals. On the other hand, in the case of the data-driven approach the dictionaries are directly learned from the data. This approach is supported by the assumption that the structure of complex natural phenomena can be more accurately extracted by learning the bases directly from the data rather than by a closed-form mathematical description. Given the example vectors, the atoms for a dictionary can either be chosen from the examples or be derived using a learning algorithm. There exist a number of dictionary learning algorithms such as the *method of optimal directions* (MOD) [75], *union of ortho-bases* [76], *generalized PCA* [77] and KSVD [62]. Among those algorithms the KSVD happens to be the most popular one due to its effectiveness.

3.2 Speaker recognition using sparse representation over exemplar dictionary

In this section, the existing approaches employing sparse representation over an exemplar dictionary for SI and SV tasks are described. Different kind of representations used for creating the exemplar dictionary are also reviewed.

3.2.1 Speaker identification

Consider a speaker identification problem involving N speakers with n_i examples being available for training the i^{th} speaker. Let $\boldsymbol{x}_{ij} \in \mathbb{R}^m$ denote the suitable vector representation for the j^{th} example of the i^{th} speaker. All training example vectors of the i^{th} speaker are combined to form a matrix $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{in_i}] \in \mathbb{R}^{m \times n_i}$. It is assumed that a test vector $\boldsymbol{y}$ belonging to the i^{th} speaker lies in the linear span of the training examples of the same speaker and hence it can be approximated as,

$$\boldsymbol{y} \approx s_{i1}\boldsymbol{x}_{i1} + s_{i2}\boldsymbol{x}_{i2} + \cdots + s_{in_i}\boldsymbol{x}_{in_i}, \tag{3.3}$$

where $\{s_{ij}\}_{j=1}^{n_i}$ are real scalars. For the identification purpose, an exemplar dictionary is created by concatenating all N speaker matrices as,

$$\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N] \in \mathbb{R}^{m \times k}, \quad K = \sum_{i=1}^{N} n_i. \tag{3.4}$$

Now $\boldsymbol{y}$ is approximated as linear combination of k columns of the dictionary $\boldsymbol{X}$ as,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{s} \tag{3.5}$$

where $\boldsymbol{s} \in \mathbb{R}^k$ is the vector of unknown coefficients. With the assumption made in Equation 3.3, $\boldsymbol{s}$ is expected to be sparse. The sparse solution $\hat{\boldsymbol{s}}$ to Equation 3.5 can be obtained by solving the following objective function,

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{s}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{s}\|_0 \leq l \tag{3.6}$$

where, l is the chosen constraint on sparsity.

Ideally all nonzero coefficients of $\hat{\boldsymbol{s}}$ should correspond to the atoms from the class of $\boldsymbol{y}$ only. But in practice due to the modeling error, noise and channel variability, the atoms other than those corresponding to the class of $\boldsymbol{y}$ will also have nonzero coefficients. The classification is done by comparing the class-wise l_1-norm of the sparse representation vector as,

$$\arg\max_i \|\delta_i(\hat{\boldsymbol{s}})\|_1, \tag{3.7}$$

or by comparing the class-wise reconstruction error as,

$$\arg\min_i \|\boldsymbol{y} - \boldsymbol{X}\delta_i(\hat{\boldsymbol{s}})\|_2^2, \tag{3.8}$$

where $\delta_i(.)$ is a window operator that sets all the coefficients in a vector to zero, except those corresponding to the i^{th} class.

3.2.2 Speaker verification

To perform SV using sparse representation, for each claim an exemplar dictionary $\boldsymbol{X}_{\mathrm{c}}$ is created by concatenating the vectors representing the i^{th} (claimed) speaker and that of a set of background (non-target) speakers as,

$$\boldsymbol{X}_{\mathrm{c}} = [\boldsymbol{X}_i | \boldsymbol{X}_{\mathrm{b}}] \tag{3.9}$$

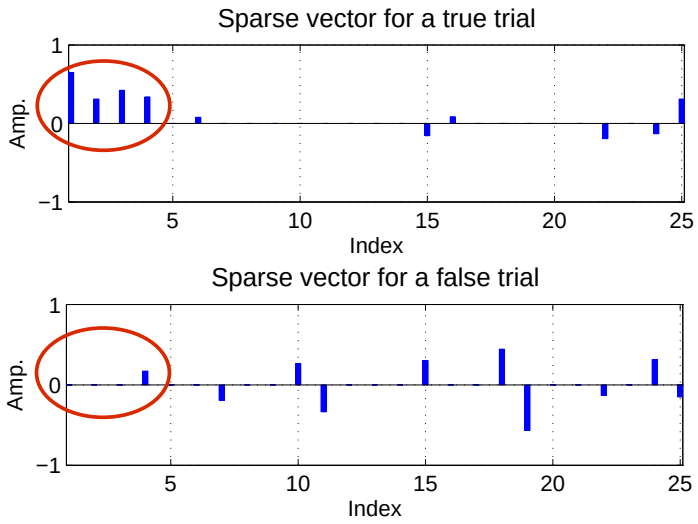


Figure 3.2: Example sparse vectors for a true trial and a false trial in case of SR based SV over a small exemplar dictionary created for the ease of display. The first 5 atoms in the dictionary are the training utterances of the claimed speaker and the rest are those of the background speakers. Note the difference between the structure of sparse codes for true and false trials.

where $\boldsymbol{X}_i$ represents the set of training example vectors of the claimed speaker and $\boldsymbol{X}_\mathrm{b}$ denotes that of the background speakers. Now, the sparse representation $\hat{\boldsymbol{s}}$ for a test vector $\boldsymbol{y}$ over $\boldsymbol{X}_\mathrm{c}$ is optimized using Equation 3.6. To illustrate the efficacy of this approach, a toy-experiment is conducted with a dictionary having 25 atoms where the first 5 atoms are the training examples of the claimed speaker and the remaining are examples of the background speakers. Figure 3.2 shows the sparse vectors corresponding to a true trial and a false trial computed over the created dictionary with a sparsity constraint of 10. Note that for the true trial, the predominant coefficients in the representation of the test vector belong to the atoms corresponding to the claimed speaker. On the other hand, for the false trial, the atoms representing the background speakers are also significantly involved in the representation. Hence, the score for verification can be computed by measuring either the sum of the magnitude of the sparse coefficients or the representation error of the test vector, separately for the target and the background speaker atoms. The typical scoring metrics explored in literature [24, 26] are as follows,

$$\text{Magnitude } l_1\text{norm (M-1)} \quad : \quad \frac{\|\delta_i(\hat{\boldsymbol{s}})\|_1}{\|\hat{\boldsymbol{s}}\|_1} \tag{3.10}$$

$$\text{Residue } \ell_2\text{-norm (M-2)} \quad : \quad \frac{\|\boldsymbol{y} - \boldsymbol{X}_\mathrm{c}\delta_\mathrm{b}(\hat{\boldsymbol{s}})\|_2}{\|\boldsymbol{y} - \boldsymbol{X}_\mathrm{c}\delta_i(\hat{\boldsymbol{s}})\|_2} \tag{3.11}$$

where $\delta(.)$ is the coefficient selection operator defined in Section 3.2.1. The sparse representation over exemplar dictionary based SV system is referred to as XD-SR SV system in this work.

3.2.3 Speaker representations

The two kinds of speaker representations predominantly used in SR over exemplar dictionary based SV systems are GMM supervectors [24,25] and i-vectors [26,27]. In typical SV systems, the size of GMM supervectors used is of the order of $50,000$. At present, it is not possible to collect that many speaker utterances from the publicly available SV databases. As a result, with the GMM supervectors as the speaker representation, the exemplar dictionary formed becomes *undercomplete*, i.e., the number of examples being less than the dimensionality of the representation. Although undercomplete dictionaries have been used for sparse coding in some works [78,79], it is a common assumption that the dictionary needs to be *overcomplete* in the SR framework.

On account of significant correlation among the elements, the intrinsic dimensionality of the GMM supervector happens to be very much smaller (typically in the range of 400-800) compared to its actual dimensionality. This fact is evident form the success of the low-rank modeling approaches like JFA and i-vector. So we argue that even an undercomplete dictionary created using supervectors and having a few thousand columns will be redundant enough to produce a sparse representation of the speakers supervectors. Alternatively, one can use the low-dimensional i-vectors instead of the supervectors to obtain an overcomplete dictionary. We have built SV systems following both of these approaches on same dataset as contrast systems and compared their performances.

Further, in context of SR based face video verification [80], it is shown that with the mean shifting (centering) of the GMM supervectors the mutual coherence of the dictionary is improved and it results in an improved classification performance. The centered GMM supervector, $\boldsymbol{y'}$ is derived as,

$$\boldsymbol{y'} = \boldsymbol{y} - \boldsymbol{\mu} \tag{3.12}$$

where, $\boldsymbol{y}$ is the GMM supervector for the speaker utterance and $\boldsymbol{\mu}$ is the speaker-independent UBM mean supervector. Motivated by the above stated facts, in this work, centered GMM supervectors are used for representing the speaker utterances.

3.3 Speaker verification with sparse representation over a learned dictionary

In this section we propose a novel SV approach employing sparse representation over a learned dictionary. First the issues with the exemplar dictionary based approach are highlighted which provided the motivation for this work.

3.3.1 Motivation

Despite exhibiting a competitive performance, the SR over exemplar dictionary based SV approach suffers from the following shortcomings:

- For the cases where the number of training examples available for a speaker is very small, the assumption that a target can be expressed as a linear combination of the training examples may not hold well enough.
- A single selection of the background speaker set for creating an exemplar dictionary may not be optimal for all the speakers while making such selections for each of the claimed speakers is a non-trivial task.
- The restricted isometric property (RIP) [81] and a high mutual coherence [71] are desired for a dictionary to produce sparse representation. These attributes are difficult to achieve in case of an exemplar dictionary.

To address the above mentioned issues, one solution is to learn a dictionary using examples from a large number of speakers such that all target speakers can be represented in a sparse manner in a unified space. Let $\boldsymbol{D}$ denotes the learned dictionary of K atoms which spans a space $Span\boldsymbol{D}$. Assume that an i^{th} target speaker lies in an unknown subspace formed by a set of M_i atoms $\{\boldsymbol{d}_{i_m}\}_{m=1}^{M_i} \triangleq \boldsymbol{D}_i \subset \boldsymbol{D}$. Thus an example of the i^{th} speaker $\boldsymbol{y}_i$ can be approximated as,

$$\boldsymbol{y}_i \approx \sum_{m=1}^{M_i} s_{i_m} \boldsymbol{d}_{i_m}, \quad s_{i_m} \in \mathbb{R}. \tag{3.13}$$

On representing over the dictionary $\boldsymbol{D}$, $\boldsymbol{y}_i$ finds a sparse representation as,

$$\boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{s}_i \quad \text{such that} \quad \|\boldsymbol{s}_i\|_0 \leq l, \tag{3.14}$$

where l is the chosen constraint on sparsity. The sparse solution $\hat{\boldsymbol{s}}_i$ can be obtained by optimizing the objective function given in Equation 3.6. It is argued that, the atoms of $\boldsymbol{D}$ that are not involved in representing a target speaker vector act as an optimal set of background speakers. An example

$\boldsymbol{y}_j$ form another speaker j finds a representation in its own subspace $\boldsymbol{D}_j \subset \boldsymbol{D}$. It is desired that on seeking the sparse representation of $\boldsymbol{y}_j$ over $\boldsymbol{D}_i$ would result in high representation error compared to that over $\boldsymbol{D}_j$, i.e.,

$$\|\boldsymbol{y}_j - \boldsymbol{D}_i \hat{\boldsymbol{s}}'_j\|_2^2 > \|\boldsymbol{y}_j - \boldsymbol{D}_j \hat{\boldsymbol{s}}_j\|_2^2 \tag{3.15}$$

The above inequality ensures that for any two target speakers, the sets of atoms involved in their sparse representations over the dictionary $\boldsymbol{D}$ are maximally disjoint.

3.3.2 Dictionary learning

The aim of the dictionary learning process is to create a redundant dictionary that produces a sparse representation for a large pool of speakers. In this work, we have explored both simple and discriminative approaches for learning the dictionary. For learning a simple dictionary the KSVD algorithm is used and a supervised variant of KSVD is used for learning the discriminative one. In the following, these dictionary learning algorithms are presented and some issues related to the creation of dictionaries on supervectors are discussed.

3.3.2.1 KSVD learned dictionary

Given a set of examples $\boldsymbol{X} \equiv \{\boldsymbol{x}_i\}_{i=1}^N$, the objective function for learning a dictionary for sparse representation can be expressed as,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}} = \arg\min_{\boldsymbol{D},\boldsymbol{U}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{U}\|_2^2 \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i \tag{3.16}$$

where $\boldsymbol{D}$ is the dictionary of k atoms, $\boldsymbol{U} \equiv \{\boldsymbol{u}_i\}_{i=1}^N$ is the matrix of sparse vectors and l is the chosen sparsity constraint.

The KSVD algorithm [62] solves the optimization problem given in Equation 3.16 using an iterative process having two stages: the sparse coding stage and the dictionary update stage. In the sparse coding stage, a pursuit method such as OMP [68] is used to compute the sparse representation of the given set of examples over the current dictionary. In the dictionary update stage, the data vectors associated with each of the atoms are determined. For each atom, the residue of the data vectors over the dictionary excluding that atom is computed. The chosen atom is then updated with the topmost principal component of the residue computed using SVD. The detailed steps involved in the KSVD algorithm are presented in Appendix B.

As discussed earlier in Section 3.2.3, the dictionaries created with supervectors as speaker

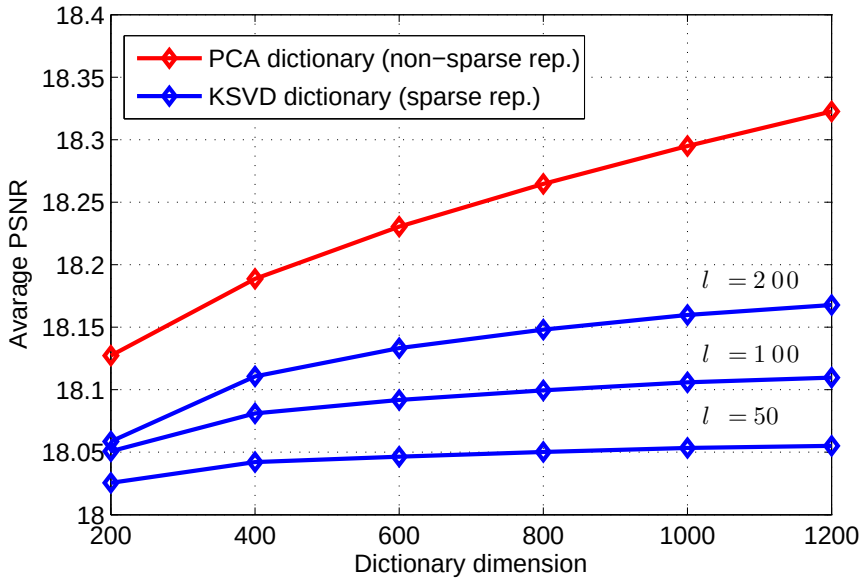


Figure 3.3: Quality of reconstruction (in PSNR) achieved with undercomplete dictionaries of different size for varying sparsity (l) in representation. For contrast, the quality of the non-sparse representation with corresponding size PCA based projections are also shown.

representations turn out to be highly *undercomplete*. This motivated us to assess the quality of reconstruction of the sparse representation produced by such an undercomplete dictionary learned using KSVD. For the same, dictionaries of varying size are learned using a set of 3000 supervectors. A different set of 1000 supervectors is then sparse coded over the learned dictionaries using the OMP algorithm with varying sparsity constraints. Figure 3.3 shows the quality of reconstruction in terms of the average peak signal-to-noise ratio (PSNR) for different sizes of the dictionary as well as for varying sparsity. For contrast purpose, the quality of reconstructions of the same set of vectors with the PCA based representation for varying projection dimensions are also shown in the figure. Note that, there is only a slight degradation in the quality of reconstruction with SR over the undercomplete dictionary (1% relative on an average, for a sparsity value of 100) compared to that with the non-sparse representations obtained using PCA. These results clarify the apprehension that a meaningful sparse representation can not be obtained with an undercomplete dictionary.

3.3.2.2 Discriminative learned dictionary

The KSVD algorithm is primarily developed for the sparse representation of data for reconstruction purpose. For classification tasks, a number of its discriminative variants which use labeled training data have been proposed. These algorithms are intended to provide better class

separability in the sparse representation domain. The supervised KSVD (S-KSVD) [79] algorithm uses *class supervised simultaneous* OMP (CSSOMP) in the sparse coding stage of the dictionary learning process which differs from OMP in two aspects: (i) CSSOMP uses the same set of atoms from the dictionary to represent all examples from a given class and so attempts to extract the common internal structure of that class whereas OMP treats each example independently, and (ii) in addition to the minimum error criterion used in OMP, CSSOMP uses a discrimination measure also which increases the separability among classes.

Given a set of labeled data vectors $\boldsymbol{X} \equiv \{\{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}\}_{i=1}^{N}$, the optimization of the discriminant dictionary $\boldsymbol{D}$ using the S-KSVD algorithm is represented as,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}} = \arg\max_{\boldsymbol{D},\boldsymbol{U}} \left\{ \theta . J\left(\boldsymbol{X}\right) - \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{U}\|_2^2 \right\} \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i \tag{3.17}$$

The function $J(.)$ represents the discriminant measure defined as $:= \frac{trace(\boldsymbol{B})}{trace(\boldsymbol{W})}$ where $\boldsymbol{B}$ and $\boldsymbol{W}$ are the *between-class* and the *within-class* covariance matrices of the learning data $\boldsymbol{X}$, respectively. $\boldsymbol{U} \equiv \{\{\boldsymbol{u}_{ij}\}_{j=1}^{n_i}\}_{i=1}^{N}$ is the matrix of sparse vectors corresponding to the data vectors $\boldsymbol{X}$. l is the sparsity constraint and θ is a parameter controlling the trade-off between discriminative and reconstructive terms in the learning criterion. The steps involved in the CSSOMP and the SKSVD algorithms are given in Appendix A and Appendix B, respectively.

3.3.3 Verification process

In case of the learned dictionary based system, the metrics given in Section 3.2.2 can not be used directly since the dictionary does not have speaker label attached to its atoms. To address this, one could determine the atoms associated with a claimed speaker by sparse coding the corresponding training examples using the learned dictionary. By considering the indices of the atoms corresponding to the non-zero coefficients in the sparse codes of training data as the target speaker label, one can use the metric in Equation 3.10 or Equation 3.11 to compute the verification score.

In contrast to the exemplar dictionary case, we could define another metric for verification purpose based on the similarity between the sparse codes of the test vector and the training examples of the claimed speaker computed with respect to the learned dictionary. To explain that, a toy experiment is conducted using a dictionary of 25 atoms learned using KSVD. Figure 3.4 shows the sparse representation vectors for a training example of a target speaker along with that of the test examples corresponding to a true claim and a false claim, computed over the created

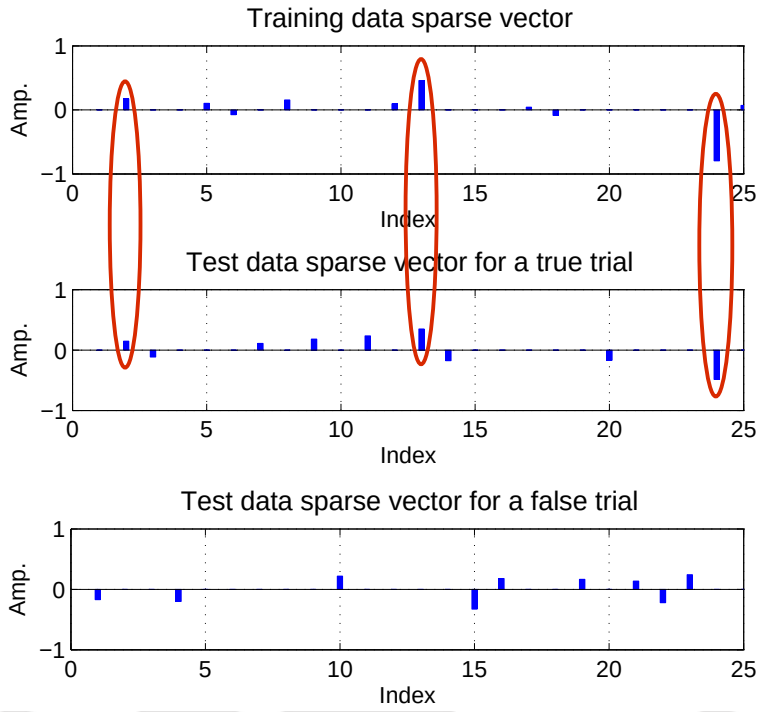


Figure 3.4: Example sparse vectors for a claimed speaker's training data and those of a true trial and a false trial over a learned dictionary having 25 atoms (for ease of display). Note the similarity between the true-trial test vector with the train vector and the lack of the same for the false-trial test vector.

dictionary with a sparsity of 10. On comparing the sparse vectors corresponding to the target and that of the true claim, we note that a significant number of dominant coefficients (encircled in the figure) in both the representations correspond to the same atoms in the dictionary. On the other hand, for the false claim no such match is observed. To exploit this behavior in the sparse domain, a simple cosine distance scoring (CDS) metric can be used for computing the similarity score as,

$$\text{Score} = \frac{< \hat{\boldsymbol{s}}_{clm}. \hat{\boldsymbol{s}}_{tst} >}{\|\hat{\boldsymbol{s}}_{clm}\| \, \|\hat{\boldsymbol{s}}_{tst}\|} \tag{3.18}$$

where $\hat{\boldsymbol{s}}_{clm}$ and $\hat{\boldsymbol{s}}_{tst}$ represent the sparse representations of the training and the test utterances, respectively. In CDS, apart from the information about the indices of atoms, their relative significance in representing the given speaker is also used which makes it superior to the earlier defined metrics. The SV systems developed in this work employing SR over learned and discriminatively learned dictionaries with CDS are referred to as LD-SR and DLD-SR systems, respectively.

3.3.4 Contrast with the i-vector based SV system

The factor analysis based i-vector model for a given centered GMM supervector $\boldsymbol{y}'$ is given as,

$$\boldsymbol{y}' = \boldsymbol{T}\boldsymbol{w}, \tag{3.19}$$

where $\boldsymbol{w}$ is the i-vector representation and $\boldsymbol{T}$ is the *total variability matrix* (T-matrix). On comparing Equation 3.19 with the model for the proposed LD-SR system given in Equation 3.14, one may find these models being quite similar but they are actually different in certain aspects. The fundamental difference lies in the nature of the projections obtained. The projection matrix (T-matrix) used to compute the i-vectors constitutes the eigen vectors of the total-variability subspace estimated using the development data and the i-vectors are derived using a Gaussian prior distribution. Though it is a known fact that the estimated i-vectors follow a non-Gaussian heavy tailed distribution, these are non-sparse in nature. On the other hand, the learning process for the dictionary used in the LD-SR system involves clustering with a constraint on sparsity followed by data compaction. As a result, the dictionary is able to produce a meaningful sparse representation when it is enforced. Another important attribute of the dictionary based approach is its ability to select the appropriate set of atoms for a given target unlike the use of a fixed set of bases in case of the i-vector approach. This attribute of the dictionary based approach turns out to be handy in addressing any possible mismatch between the development and the target data.

3.4 Experimental setup

The performances of the proposed SR over learned dictionary based systems (LD-SR and DLD-SR) are primarily contrasted with that of the SR over exemplar dictionary based SV system (XD-SR). Exemplar dictionaries created using both supervector and i-vector representations are explored. As the proposed SV systems employ CDS metric, for direct comparison purpose two CDS based systems, the one using supervector and the other using i-vector as utterance representations, are also developed and evaluated. The following subsections provide the details of various systems developed, the dataset and the performance measures used for evaluation.

3.4.1 Dataset and system parameters

The experiments are performed using the NIST 2003 SRE database. It contains speech data of 356 target speakers collected over cellular phone network. The evaluation of the system is done as

per the NIST 2003 SRE evaluation plan for primary task [82]. This experimental setup contains 24,981 trials for verification task including true and false trials.

All speech data used is sampled at 8 kHz with 16 bits/sample resolution and analyzed using a Hamming window of length 20 ms and with a frame shift of 10 ms. To remove the silence portions from speech data, an energy based voice activity detector with a threshold equal to 0.06 times the average energy of the utterance is employed. Standard MFCC features are computed using 22-channel mel-filterbank and a pre-emphasis factor of 0.97. Static MFFCs of 13 dimensions are appended with their first and second derivatives and used as 39 dimensional acoustic features for all the systems. The cepstral mean subtraction and variance normalization are also performed on the acoustic feature vectors.

The Switchboard Cellular Part 2 corpus is used as the development data for all the systems reported in this chapter. A gender-independent UBM model of 1024 Gaussian mixtures created using approximately 10 hours of the development speech data is used for all the systems. The GMM supervectors are created by adapting only the mean parameters of the UBM using MAP approach with the speaker specific data. The total variability matrix of 400 columns for the i-vector based system and the dictionary of 400 atoms for the proposed SR-SV systems are learned using 1872 speech utterances taken from the development database. In the learning of the discriminative dictionary, the trade-off parameter θ is set to 0.7. From the development database, 400 imposter speaker utterances are derived and used for creating the dictionary for the SRC system with exemplar dictionary. A simplified JFA model without residual factors [83] is developed using the code available at [84] and used for normalizing the session/channel variabilities in GMM supervectors. Using the development data, 300 dimensional speaker subspace matrix and 100 dimensional channel subspace matrix are learned for the JFA model. The LDA and WCCN matrices are created using the same development data which is used for learning the T-matrix and the dictionaries. The LDA for the i-vector system projects to 250 dimensional space whereas for the SRC based system it projects to 375 dimensional space. All the above mentioned parameters are chosen out of experimentation.

3.4.2 Performance measures

The performance of the SV systems are primarily evaluated using the EER. In addition, the minimum detection cost as defined for the NIST 2003 SRE is also used as a secondary performance

measure which is defined as follows. Given the miss probability $P_{Miss|Tar}$ and the false-alarm probability $P_{FA|Nontar}$, computed by thresholding the verification scores, the detection cost function C_{DET} is defined as,

$$C_{DET} = C_{miss} \times P_{Miss|Tar} \times P_{Tar} + C_{FA} \times P_{FA|Nontar} \times (1 - P_{Tar}) \tag{3.20}$$

where C_{FA} and C_{Miss} denote the costs associated with false alarms and misses, respectively. The parameter P_{Tar} is the prior probability of the target trial. In NIST 2003 SRE, the values of C_{FA}, C_{Miss} and P_{Tar} are set to 1, 10 and 0.1, respectively. A set of C_{DET} values are derived by varying the threshold used for computing the miss and false-alarm probabilities and using the above given parameters. The minimum of these values is referred to as min-$C_{DET\text{-}03}$ and is used as a performance measure for the experiments reported in this chapter.

3.5 Experimental results and discussions

In this section the results of the various experiments performed with the proposed SR based and the contras SV systems on the NIST 2003 SRE dataset are presented and discussed. The tuning of the different parameters of the proposed system is provided in the beginning. It is followed by the discussion on the use of the T-matrix of the i-vector system as the dictionary for the SR based SV system for the purpose of highlighting the importance of specialized algorithms like KSVD for learning dictionary for SR. The performances of the proposed as well as the contrast SV systems with and without session/channel compensation applied are also presented and analyzed.

3.5.1 Tuning of the LD-SR system parameters

In the KSVD learned dictionary based LD-SR system, there are three main parameters which required tuning : (i) the number of atoms in the dictionary, (ii) the number of atoms selected while learning the dictionary (dictionary sparsity) and (iii) the number of atoms selected while representation of target data (decomposition sparsity). The significance of the first parameter is obvious. To be consistent with the i-vector dimension reported in literature, the number of atoms in the learned dictionaries are chosen to be 400. For explaining the significance of the other two parameters, recall that the KSVD dictionary learning process involves two stages: the sparse representation of the development data and the dictionary update. Unlike the sparse representation of the unseen training as well as test data in an SV system, the dictionary learning process involves

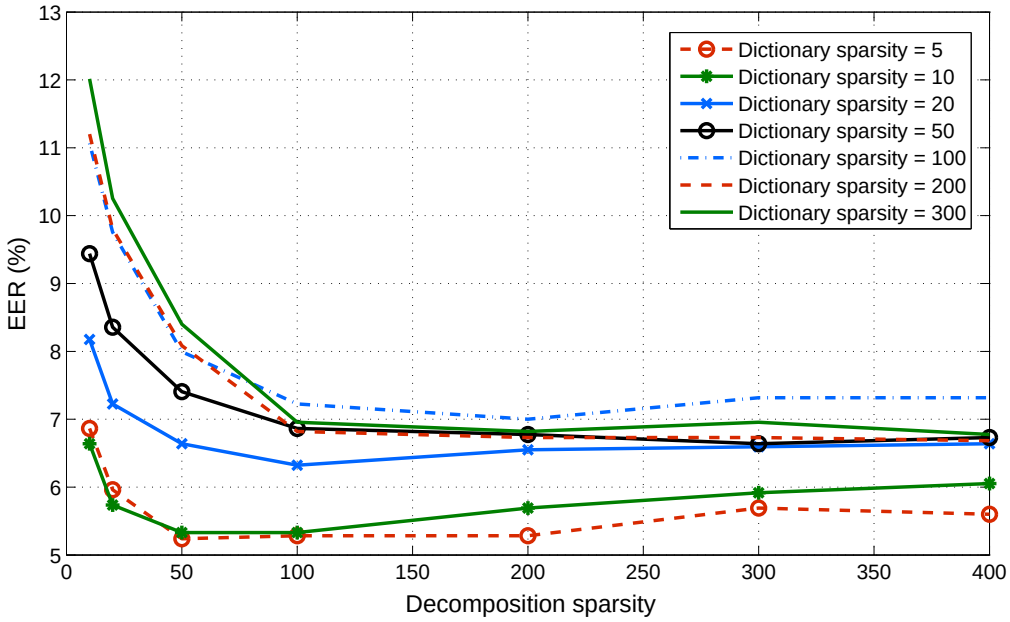


Figure 3.5: Tuning of number of atoms selected while learning the dictionary (dictionary sparsity) and the number of atoms selected for representation (decomposition sparsity) in case of the LD-SR system.

the sparse representation of the seen development data. Thus there is a scope of tuning these two parameters for optimal system performance. The performances of the system obtained while tuning the dictionary sparsity and the decomposition sparsity are shown in Figure 3.5. Note that the best system performance is obtained for the selection of 5 atoms while learning the dictionary and 50 atoms while representing (decomposing) the training and test supervectors. We have used these parameter values for the LD-SR and DLD-SR systems thorough out the work unless specified otherwise.

3.5.2 Exploration of the total variability dictionary for LD-SR system

As discussed in Section 3.3.4, both the LD-SR and the i-vector CDS systems use data-dependent learned dictionaries to compute the representations. This motivated us to explore the possible use of the total variability matrix of the i-vector approach as a dictionary (*T-dictionary*) for the LD-SR system. The resultant system is referred to as T-LD-SR system. The performance of the T-LD-SR system with a T-dictionary of 400 atoms is evaluated with varying number of atoms being selected for representations. For comparison purpose, the experiment is repeated with an LD-SR system with KSVD dictionary of 400 atoms. These performances are shown in Figure 3.6 along with the performance of a 400 dimensional i-vector CDS system for contrast purpose. It can be noted that

the T-LD-SR system gives very poor performance when small number of atoms (say 10 atoms) are selected. With increasing number of atoms, the performance improves significantly and for all 400 atoms being selected, it matches with that of the 400-dimensional i-vector based system. On comparing with the LD-SR system, we note that the T-LD-SR system gives slightly better performance for more than 200 atoms selected, but for smaller number of atoms it is found to result in significantly degraded performance. With the requirement of more number of atoms in case of the T-dictionary, it will be interesting to explore how a T-LD-SR system performs with a bigger size dictionary (i.e., having more number of columns) and with relaxation in sparsity (i.e., more atoms being used for sparse representation). These aspects are explored in the following.

3.5.2.1 Effect of size of the dictionary for T-LD-SR system

To explore the effect of size of the dictionary in the T-LD-SR system performance, dictionaries of 200, 300, 400, 600, 700 and 900 columns are created. The T-LD-SR systems with these dictionaries are evaluated for three different numbers of atoms selected for representations viz. 50, 200 and 400. The performance of these systems are shown in Figure 3.7 along with that of the corresponding i-vector and LD-SR systems. For all the systems considered, the performance is found to degrade consistently for dictionary sizes beyond 400 atoms. For the T-LD-SR system with higher number of atoms selected for representations, the performance is found to be improving and becoming comparable to that of the i-vector based system closely. It is interesting to note that for all the three types of systems considered, the best performances correspond to dictionary sizes between 300-400.

3.6 Performance comparison

The performances of the proposed and the contrast systems evaluated on the NIST 2003 SRE dataset are given in Table 3.1. The CDS based systems with the GMM mean shifted supervectors and the i-vectors as speaker representations have resulted in EERs of 8.42 % and 4.21 %, respectively. The low-dimensional i-vector based system has already reported to significantly outperform the much larger dimensional GMM mean shifted supervector based system. The SRC over exemplar dictionary based systems result in an EER of 6.50 % and 6.78 % for the GMM mean supervector and the i-vector representations, respectively. Note that these systems have resulted in lower performance than the i-vector CDS based system and this trend is consistent

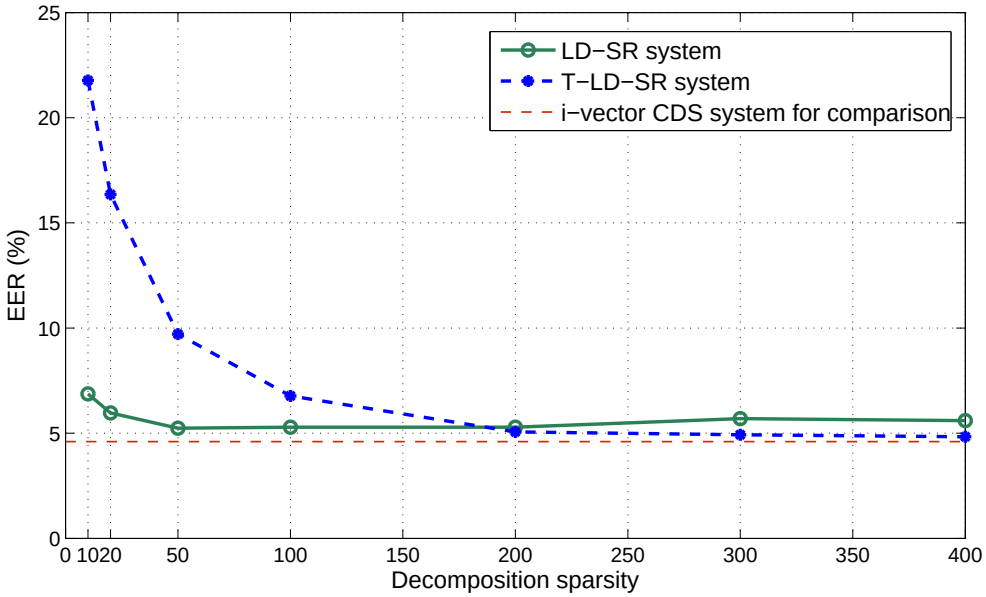


Figure 3.6: Effect of number of atoms selected for sparse representation in the cases of the LD-SR and the T-LD-SR systems.

with those reported in literature [24, 26]. The three different variants of the proposed SR based system i.e., with T-dictionary (T-LD-SR), with KSVD dictionary (LD-SR) and with S-KSVD dictionary (DLD-SR), are evaluated and these systems have resulted in EERs of 5.05 %, 5.23 % and 2.89 %, respectively. Note that the reported performance of the T-LD-SR system corresponds to a decomposition sparsity value equal to 200 in contrast to the value of 50 being used for the cases of LD-SR and DLD-SR systems. Thus the slightly better performance noted in the case of the T-LD-SR system compared to that of the LD-SR system is at the cost of a higher complexity. It is to note that all the versions of the LD-SR systems have significantly outperformed both the versions of the XD-SR system, thus emphasizing the effectiveness of learned dictionaries for the SR based SV task. On comparing further, it can be noted that both the T-LD-SR and the LD-SR systems have performed somewhat inferior to the i-vector CDS system. On the other hand, the DLD-SR system has outperformed the i-vector CDS system significantly. It is also noted that the performances of all the systems in terms of min-$C_{\text{DET-03}}$ follow the same trend as that in case of EER.

As discussed in Section 3.3.4, there are some obvious similarities between the proposed LD-SR and the i-vector CDS systems those can be noted by comparing Equation 3.14 with Equation 3.19. The matrices $\boldsymbol{D}$ and $\boldsymbol{T}$ have the same size and the projections $\boldsymbol{x}$ and $\boldsymbol{w}$ of the GMM mean shifted

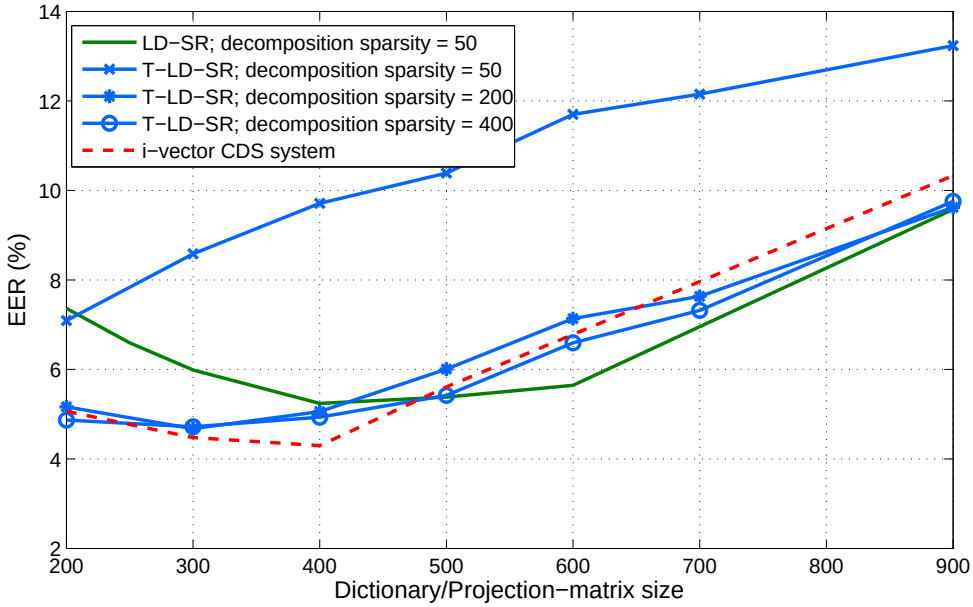


Figure 3.7: Effect of the use of varying size dictionary for the T-LD-SR system. The performances of corresponding size dictionary/projection matrix for the LD-SR and the i-vector CDS systems are also shown for contrast purpose.

Table 3.1: Performances of various proposed and contrast speaker verification systems on NIST 2003 SRE dataset.

	SV system type	Representation	% EER	min-$C_{\text{DET-03}}$
Contrast	CDS	supervector	8.42	0.161
		i-vector	4.21	0.072
	XD-SR	supervector	6.50	0.117
		i-vector	6.78	0.121
Proposed	T-LD-SR (Decomp. Sparsity = 200)	supervector	5.05	0.088
	LD-SR (Decomp. Sparsity = 50)		5.23	0.097
	DLD-SR (Decomp. Sparsity = 50)		**2.89**	**0.051**

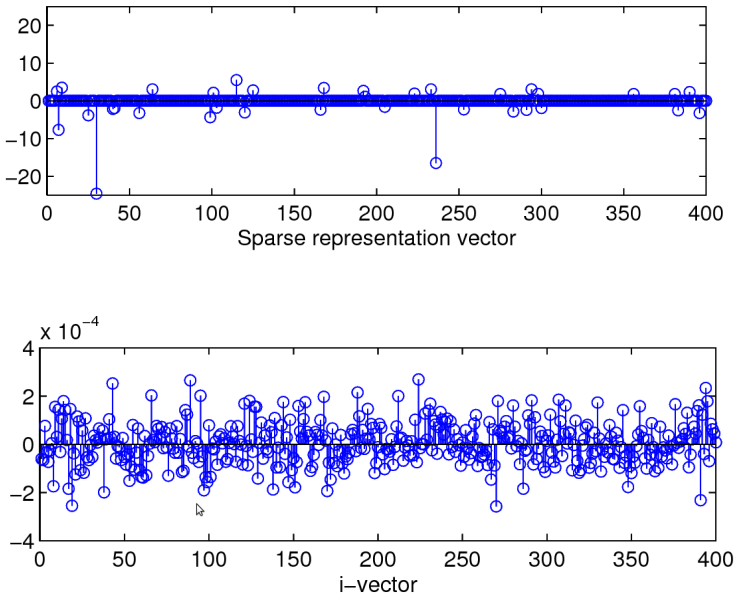


Figure 3.8: Example profiles of the learned dictionary based sparse representation and the i-vector representation, each having 400 dimensions. Note for the non-sparse nature of the i-vector representation.

supervector derived from those matrices are used for classification with the same scoring metric. The main differences between the two lie in the different criteria being used for learning those matrices and the nature of the derived projections. The Figure 3.8 shows the typical projected vectors in the case of the LD-SR and i-vector based systems. Note that the projection in case of the LD-SR system is sparse while the one in case of the i-vector based system is non-sparse. Further, we hypothesize that training of the low-rank total variability matrix $\boldsymbol{T}$ and learning of the redundant dictionary $\boldsymbol{D}$ have somewhat similar goals, i.e., to develop a more compact model for classification. On the other hand when an explicit discriminative criterion is employed in dictionary learning it significantly boosts the classification ability. As a result, proposed DLD-SR system which uses a discriminatively learned dictionary has shown significantly improved performance in comparison to the LD-SR and the i-vector CDS based ones.

3.6.1 Session/channel compensation

The session/channel compensation methods form an integral part of all modern speaker verification systems and are critical for making them effective in practical conditions. For the proposed LD-SR system, the compensation can be applied either in the supervector domain or in the sparse

Table 3.2: Performances of various proposed and contrast speaker verification systems with incorporating suitable session/channel compensation methods on NIST 2003 SRE dataset. Note that, with proper compensation both simple and discriminative dictionary based systems have outperformed the i-vector based system.

	SV system	Representation	Channel compensation	% EER	min-$C_{\text{DET-03}}$
Contrast	CDS	i-vector	LDA & WCCN	2.24	0.037
		supervector	pre-JFA	3.61	0.066
	XD-SR	i-vector	LDA & WCCN	5.42	0.102
		supervector	pre-JFA	4.01	0.069
Proposed	T-LD-SR	supervector	LDA & WCCN	3.43	0.063
	LD-SR		LDA & WCCN	3.61	0.065
	DLD-SR		LDA & WCCN	1.98	0.036
	LD-SR		pre-JFA	1.56	0.031
	DLD-SR		pre-JFA	**1.53**	**0.028**

vector domain. For performing the session/channel variability compensation in the the supervector domain the simplified JFA approach as described in Section 2.3.1 is used. Once the speaker and channel factors are estimated for a given utterance using the JFA process, the session/channel compensated GMM supervector is computed by multiplying the corresponding speaker factor vector with the speaker subspace matrix. This compensated GMM supervector is used for representing the speaker utterance in the subsequent sparse coding stage. This approach of session/channel compensation through JFA based processing of the supervectors prior to the sparse coding is referred to as 'pre-JFA' in this work. For applying session/channel compensation in the sparse vector domain, LDA followed by WCCN is used. In case of the the CDS based and the XD-SR systems LDA followed by WCCN is used for i-vectors representations and JFA is used for the supervector representations.

The performance of various SV systems evaluated on NIST 2003 SRE dataset with appropriate kind(s) of session/channel compensation applied are given in Table 3.2. On comparing Table 3.1 and Table 3.2, we note that the relative ordering of the performances of different systems considered remains the same with and without application of session/channel variability compensation except for the case of the LD-SR system. With session/channel compensation using pre-JFA, the LD-SR system is noted to perform better than the i-vector CDS system. In addition, it can be noted that the pre-JFA based compensation is more effective than the post-processing with LDA and

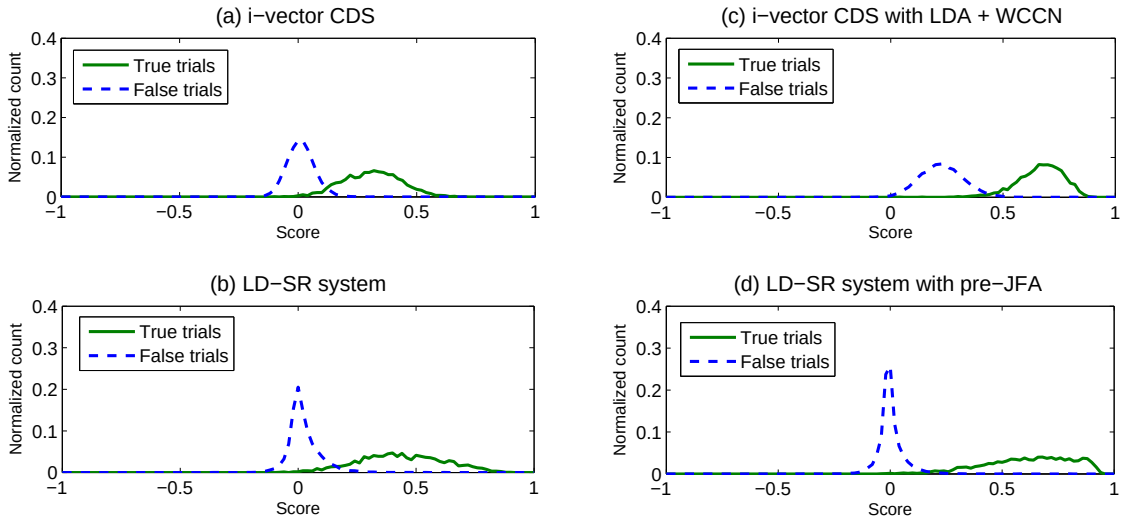


Figure 3.9: Histograms of scores generated by the i-vector CDS, LD-SR systems. Sub-figures (a)&(b) correspond to the base systems without session/channel compensation while sub-figures (c)&(d) correspond to systems with appropriate session/channel compensation.

WCCN for the LD-SR systems. The three best performing systems after session/channel variability compensation are the LD-SR, DLD-SR and the i-vector CDS based systems having EERs of 1.56 %, 1.53 % and 2.24 %, respectively.

The performances of the LD-SR system in comparison with the i-vector CDS system with and without session/channel compensation are not straight-forward. To analyze the performance of the systems further, the histograms of the scores for the true and false trials in cases of the LD-SR and the i-vector CDS based systems with and without session/channel compensation are shown in Figure 3.9. On comparing the histograms of the systems without compensation, it can be noted that the distribution of false trials scores for the LD-SR system is much narrow compared to that of the i-vector CDS system. In addition, the mean of the true trial scores are more right shifted for LD-SR system compared to that of the i-vector system. Though these two aspects of the LD-SR system are somewhat positive, the spread of the distribution of the true trial scores is much higher in case of the LD-SR system compared to that of the i-vector CDS system. As a result, with no session/channel compensation applied, the LD-SR system ended up giving poorer performance compared to the i-vector CDS system. With proper session/channel compensation employed, for the LD-SR system the mode of the distribution of the true trial scores has moved significantly away from the origin and the distribution of false trial scores has become narrower compared to the un-compensated case. For the i-vector CDS case, the separation between the distributions of the true

and false trial scores has increased, but at the same time the spread of the distribution of the false trial scores is also increased. As a result of these, with suitable session/channel compensation employed, the LD-SR system happen to provide a significantly higher performance compared to the i-vector CDS system.

3.7 Summary

In this chapter, a novel approach for speaker verification exploiting sparse representation of GMM supervectors over a learned dictionary is proposed. This work is motivated by the shortcomings of the recently proposed sparse representation over an exemplar dictionary based SV systems. Both simple and discriminative methods have been explored for learning the dictionary. The proposed system is compared to the sparse representation over exemplar dictionary based as well as the existing i-vector based SV systems on the NIST 2003 SRE dataset. The proposed LD-SR system, which is based on a learned dictionary, is found to outperform the existing exemplar dictionary based approach significantly. In particular, the discriminatively learned dictionary based DLD-SR system has shown a large margin of improvement and thus resulting in a performance that is even better than the i-vector CDS based contrast system. To analyze the system performance with explicit session/channel compensation, the JFA preprocessing is used along with the proposed systems and LDA followed by WCCN is used for the i-vector CDS based contrast system. Unlike the trend noted for the uncompensated case, with JFA preprocessing even the LD-SR system is found to significantly outperform the i-vector CDS based contrast system. With session/channel compensation employed, the DLD-SR system happens to be the best performing one with a relative improvement of 31.6 % in terms of EER and 24.3 % in terms of minimum detection cost over the i-vector CDS based system. These performance improvements are attributed to the exploitation of discrimination in speaker space due to the sparseness in the representation and a better generalization due to the dictionary learning in the proposed system. These results and observations lead to the following conclusions: (i) the sparse representation of supervectors over a learned dictionary is *more sensitive* to the session/channel mismatches in comparison to the i-vector approach, and (ii) with proper compensation of these mismatches, the sparse representation over learned dictionary based approach becomes highly effective. The Fisher information criteria employed in the discriminative dictionary learning process can be interpreted as a step towards providing session/channel compensation the better performance of the DLD-SR system noted even

without explicit session/channel compensation. Though the observed improved performances in case of the proposed LD-SR systems are promising, the explicit channel compensation using pre-JFA increases the overall complexity of the system. Motivated by this, in the next chapter we present a novel SR based SV approach that incorporates the session/channel compensation in the sparse coding stage itself.

4

Joint Sparse Coding: SV System with Built-in Session/Channel Compensation

Contents

In Chapter 3, a novel approach for SV exploiting sparse representation of GMM supervectors over a learned dictionary has been presented. The effectiveness of the proposed approach was noted to depend very much on the application of a suitable session/channel compensation method. Out of the two session/channel compensation methods explored the one using JFA in the supervector domain (pre-JFA) is found to be more effective than the one performed in the sparse representation domain through post-processing with linear transforms such as LDA and WCCN. Though the pre-JFA approach is found to provide a significant improvement in the performance of the learned dictionary based LD-SR and DLD-SR SV systems, it is argued to suffer from the following shortcomings:

- The projection of the supervectors to a very low-rank speaker subspace in the JFA stage prior to the dictionary learning may lead to undesirably high mutual coherence among the atoms, which will affect the uniqueness of the sparse solution.
- The channel factors that are discarded in the JFA modeling are reported to contain some amount of speaker information [17].

In addition to the above mentioned facts, it is to note that the use of a non-sparse projection over the orthogonal speaker subspace for channel compensation is not concurrent with the SR paradigm explained in Section 3.3. The basic premise of the SR based SV approach lies in modeling of the clusters in the speaker space with the use of a redundant dictionary. With the prior projection of the supervectors to an orthogonal low-rank space in the pre-JFA, the speaker clusters may get smoothed out in the compensated data if the chosen subspace dimension happens to be much smaller than the size of the subsequent learned dictionary. As a result, the dictionary learned using the compensated data will exhibit an increased mutual coherence which will not be desirable for the uniqueness of the sparse solution. To avoid this, one needs to learn a much higher dimensional (typically 1000) speaker subspace in the pre-JFA stage that is in tune with the size of the dictionary. Learning such a big size subspace matrix would further increase the computational complexity of the overall approach, hence it is not preferred.

Further, we hypothesize that in the JFA based session compensation, the reported loss in the speaker information occurs due to the non-generalization of the fixed bases learned from the development data. Unlike that, in the SR paradigm, the target supervector is modeled using an adaptive selection of non-orthogonal bases (atoms) of a redundant dictionary. This flexibility in the selection of the bases can help to reduce the loss of the speaker information. To address

these problems, in this chapter we propose an approach that avoids the two-stage process by incorporating the session/channel compensation in the sparse coding stage itself. In the proposed approach two dictionaries, one representing the speaker space and the other representing the session/channel space, are learned. The joint sparse coding of the target supervectors over these dictionaries is performed and the sparse code corresponding to the speaker dictionary is used for the verification purpose. An algorithm for learning dictionaries for the joint sparse coding is also presented.

The proposed approach is evaluated using the NIST 2012 SRE dataset which is much bigger in size compared to the NIST 2003 SRE dataset employed for the initial work reported in Chapter 3. In addition to a larger number of speakers and trials, it also contains a number of variabilities in terms of channel type and noise conditions. Considering the fact that the NIST 2012 SRE task is more challenging compared to the 2003 one, the basic approaches proposed in Chapter 3 as well as the corresponding contrast systems are also trained and evaluated on the NIST 2012 SRE dataset. Additionally, the recently proposed i-vector PLDA system has been included to the set of contrast systems. The performance of the proposed joint sparse coding based SV system is analyzed in the noisy and low duration conditions in the test data and its computational complexity is discussed.

The remainder of this chapter is organized as follows. Section 4.1 describes the novel SV approach based on joint sparse coding over speaker-channel dictionaries and also presents an algorithm for the same. The details of the experimental setup are given in Section 4.2 and the results are presented in Section 4.3. In Section 4.4 we analyze the results and highlight the salient attributes of the proposed approach. The salient findings are summarized in Section 4.5.

4.1 Joint sparse coding over speaker-channel learned dictionaries

To incorporate the session/channel compensation in the sparse coding stage, two dictionaries $\boldsymbol{D}_{spk}$ and $\boldsymbol{D}_{chn}$ are learned to model the speaker and session/channel spaces, respectively. The given uncompensated target supervector is then sparse coded jointly over these dictionaries as,

$$\boldsymbol{y'} = [\boldsymbol{D}_{spk}|\boldsymbol{D}_{chn}]\boldsymbol{s} \quad \text{such that} \quad \|\boldsymbol{s}\|_0 \leq l \tag{4.1}$$

where $\boldsymbol{s} \equiv [\boldsymbol{s}_{spk}^T|\boldsymbol{s}_{chn}^T]^T$ constitutes the joint sparse code corresponding to two dictionaries and '|' is the horizontal concatenation operator. The sparse projection to the speaker dictionary $\boldsymbol{s}_{spk}$ is used as the session/channel compensated representation of the target.

The dictionaries are created using a labeled development dataset following a procedure that is broadly similar to the one employed for training the JFA model. At first, for each of the speakers in the development data, a centroid supervector is computed by simply averaging their multi-condition supervectors to smooth out the session/channel variations. These centroid supervectors are then used to learn the speaker dictionary using the KSVD algorithm. Supervectors from some other multi-session and multi-channel data are then sparse coded over the learned speaker dictionary and their residues are used for learning a channel dictionary. The detailed procedure for learning these dictionaries is given in Algorithm 1.

The NIST SRE evaluation datasets used in this work contains both telephone and microphone recorded speaker utterances and the test trials include cross condition testing. For more effective session/channel compensation under both the sensor conditions, separate channel dictionaries $\boldsymbol{D}_{phn}$ (for telephone) and $\boldsymbol{D}_{mic}$ (for microphone) pivoted on the speaker dictionary are created following the above mentioned procedure. These dictionaries are then concatenated to the speaker dictionary to form the final dictionary for the joint sparse coding as,

$$\boldsymbol{y'} = [\boldsymbol{D}_{spk}|\boldsymbol{D}_{phn}|\boldsymbol{D}_{mic}]\boldsymbol{s} \quad \text{such that} \quad \|\boldsymbol{x}\|_0 \leq l \tag{4.2}$$

where $\boldsymbol{s} \equiv [\boldsymbol{s}_{spk}^T|\boldsymbol{s}_{phn}^T|\boldsymbol{s}_{mic}^T]^T$ constitutes the joint sparse code corresponding to the component dictionaries. The sparse components $\boldsymbol{s}_{spk}$ corresponding to the speaker dictionary is then used for verification with CDS.

For learning the speaker dictionary both the simple and discriminative algorithms are explored. For the initial work reported in the Chapter 3, the KSVD and the S-KSVD algorithms are used for learning the simple and discriminative dictionaries, respectively and are found to be quite effective. In the S-KSVD algorithm, the Fisher discriminant of the sparse codes with respect to all atoms in the dictionary is required to be computed for selecting an atom in the sparse coding stage and as a result of this the computational complexity becomes quite high. Label constraint KSVD (LC-KSVD) [85, 86] is another discriminative dictionary learning algorithm proposed recently. The dictionary learning with LC-KSVD is found much faster than that with S-KSVD with no degradation in the performance. On account of the NIST 2012 SRE dataset being quite large in size, we have used the LC-KSVD algorithm for learning the discriminative dictionary. For the simple dictionary case, the previously described KSVD algorithm is used. The SV systems using SR over joint learned dictionary and joint discriminatively learned dictionary are referred to as JLD-

Algorithm 1 Dictionary learning for the joint sparse coding.

Given: Data in form of the centered GMM supervectors:

(i) Multi-condition data from N speakers for training speaker dictionary: $\boldsymbol{X} \equiv \{\{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}\}_{i=1}^{N}$, where n_i is the number of utterances available for the i^{th} speaker.

(ii) Multi-condition dataset, $\boldsymbol{Y} \equiv \{\boldsymbol{y}_i\}_{i=1}^{n_Y}$ for training the channel dictionary. Overlap between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is allowed.

Step 1: Compute the speaker centroids by finding the sample-mean of the supervectors for each speaker: $\boldsymbol{M} \equiv \{m_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\boldsymbol{x}_{ij}\}_{i=1}^{N}$

Step 2: Train the speaker dictionary $\boldsymbol{D}_{spk}$ consisting of p atoms using KSVD with the constraint on sparsity set as l_{spk},

$$\boldsymbol{D}_{spk} = \arg\min_{\boldsymbol{D}} \left\{ \|\boldsymbol{M} - \boldsymbol{D}\boldsymbol{U}\|_2^2 \right\} \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l_{spk} \quad \forall i$$

where $\boldsymbol{U} \equiv \{\boldsymbol{u}_i\}_{i=1}^{N}$ is the matrix of sparse vectors

Step 3: Sparse code the channel dataset $\boldsymbol{Y}$ with the speaker dictionary using OMP,

$$\boldsymbol{s}_i = \arg\min_{\boldsymbol{s}} \|\boldsymbol{y}_i - \boldsymbol{D}_{spk}\boldsymbol{s}\|_2^2 \quad \text{such that} \quad \|\boldsymbol{s}\|_0 \leq l_{spk}; \quad \forall i$$

Step 4: Find the residue data representing the session/channel variations as,

$$\boldsymbol{R}_{chn} = \boldsymbol{Y} - \boldsymbol{D}_{spk}\boldsymbol{S}$$

where $\boldsymbol{S} \equiv \{\boldsymbol{s}_i\}_{i=1}^{n_Y}$ is the matrix of sparse codes obtained in **Step 3**

Step 5: Train the channel dictionary $\boldsymbol{D}_{chn}$ consisting of q atoms on the residue data using KSVD with the constraint on sparsity set as l_{chn},

$$\boldsymbol{D}_{chn} = \arg\min_{\boldsymbol{D}} \left\{ \|\boldsymbol{R}_{chn} - \boldsymbol{D}\boldsymbol{V}\|_2^2 \right\} \quad \text{such that} \quad \|\boldsymbol{v}_i\|_0 \leq l_{phn} \quad \forall i$$

Step 6: Create the final dictionary for the joint sparse coding by concatenating the speaker and the channel dictionaries as: $\boldsymbol{D} = [\boldsymbol{D}_{spk}|\boldsymbol{D}_{chn}]$

SR and JDLD-SR systems, respectively. The LC-KSVD algorithm is described in the following subsection. A block diagram showing the various processes involved in the dictionary training and sparse coding stages of the proposed joint dictionary based SV system is shown in Figure!4.1. Typically thousands of supervectors are used to learne the speaker and channel dictionary and the typical values for the parameters p and q are 1000 and 200, respectively.

4.1.1 Discriminative dictionary learning with label constraint KSVD

In LC-KSVD, a label consistent constraint called discriminative sparse-code error is jointly minimized with the reconstruction error along with a constraint on sparsity. Using this combined

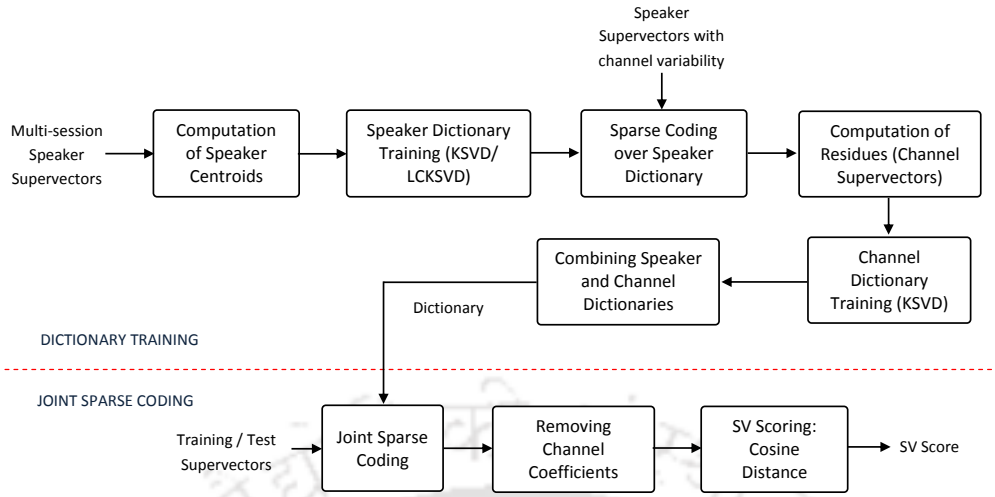


Figure 4.1: Block diagram showing the dictionary learning and sparse coding for the joint speaker-channel dictionary based SV systems.

optimization, the algorithm learns the dictionary and a linear transformation that maps the raw sparse codes to a more discriminative ones. The objective function used for the learning process is as follows,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{R}} = \underset{\boldsymbol{D},\boldsymbol{U},\boldsymbol{R}}{\arg\min} \left\{ \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{U}\|_2^2 + \alpha \|\boldsymbol{S} - \boldsymbol{R}\boldsymbol{U}\|_2^2 \right\} \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i \tag{4.3}$$

where $\boldsymbol{D}$ is the learned dictionary, $\boldsymbol{X}$ is the set of training data vectors and $\boldsymbol{U}$ is the set of corresponding sparse codes. α is the regularization parameter and the matrix $\boldsymbol{S}$ contains the required discriminative sparse codes created using the class label information of the data vectors. $\boldsymbol{R}$ is a linear transformation matrix that maps the sparse codes of the training targets to a more discriminative ones. For the optimization purpose, the two components of Equation 4.3 are rearranged as below,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{R}} = \underset{\boldsymbol{D},\boldsymbol{U},\boldsymbol{R}}{\arg\min} \left\| \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\alpha}\boldsymbol{S} \end{pmatrix} - \begin{pmatrix} \boldsymbol{D} \\ \sqrt{\alpha}\boldsymbol{R} \end{pmatrix} \boldsymbol{U} \right\|_2^2 \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i. \tag{4.4}$$

Now, Equation 4.4 can be solved using the KSVD algorithm as described in Section 3.3.2.1.

4.2 Experimental setup

The performances of the proposed joint sparse coding based JLD-SR and JDLD-SR systems are primarily contrasted with that of the state-of-the-art Gaussian PLDA based i-vector system [19]

Table 4.1: Gender-wise breakup of the total number of speakers in the NIST 2012 SRE training dataset. The number of speakers in each of the gender category having telephone and microphone recorded data are also provided.

	Male	Female	Total
Total number of speakers	770	1161	1931
No. of speakers having telephone data	770	1161	1931
No. of speakers having microphone data	335	423	758

which is described in Section 2.4.3. As the proposed systems employ CDS metric, for direct comparison purpose a CDS based i-vector system is also developed and evaluated. Additionally, the XD-SR system is also evaluated to provide contrast to the proposed systems. As there is a change in data set used for evaluation, the LD-SR and DLD-SR systems presented in the Chapter 3 are re-evaluated. All the contrast systems are employed with suitable session/channel compensation techniques while no explicit compensation is used for the proposed JLD-SR and JDLD-SR systems. The following subsections provide the details of various systems developed, the dataset and the performance measures used for the evaluation.

4.2.1 Database

The experimental evaluation of the proposed approaches is done on the NIST 2012 SRE database [87] which contains 770 male and 1161 female speakers. The training data for the target speakers are derived from multiple recording sessions and the number of segments available across the speakers vary from 1 to 240. The training set contains telephone recorded phone conversations, microphone recorded phone conversations and microphone recorded interview recordings. The gender-wise and the sensor-wise distributions of speakers in the training dataset are given in Table 4.1. Note that the microphone recorded data is available for about half of the speakers only in the training dataset.

The test dataset contains 68,954 speech segments having a duration of 30/100/300 seconds. These segments are then used to derive 1.38 million verification trials defined for the primary task in the NIST 2012 SRE that is used for this study. In 2012 SRE, the knowledge of all targets is allowed in computing the detection score for each of the trials. To examine the effect of this protocol on the performance of the system, the test segments from unknown speakers (i. e., those which are not modeled) are also included in the test dataset. As a result, the primary task comprises

Table 4.2: Distribution of the NIST 2012 SRE test data segments across five different evaluation conditions along with their data duration-wise breakup. The values given in bracket indicate the number of trials in each cases.

Evaluation condition		Number of test data segments (# trials)			
		30 sec.	100 sec.	300 sec.	All
TC-1	Microphone: clean	9883 (48731)	7811 (38304)	5168 (24334)	22862 (111369)
TC-2	Telephone: clean	4369 (187841)	4208 (181909)	4369 (187841)	12946 (557591)
TC-3	Microphone: noisy	-	4311 (18500)	13259 (54431)	17570 (72931)
TC-4	Telephone: noisy	-	-	10095 (423891)	10095 (423891)
TC-5	Telephone: noisy environment	1860 (73073)	1761 (69675)	1860 (73073)	5481 (215821)

of true trials and *known* as well as *unknown* false trials. The test dataset is further split into five subsets based on the type of the sensor, noise and recording environment conditions. The distribution of the test segments and the corresponding number of trials in the primary task are given in Table 4.2.

For the system development, speaker utterance segments of 3-5 minutes duration derived from NIST 2006, 2008 and 2010 SRE datasets are used. Among those utterances, approximately $19k$ ($7k$ male and $12k$ female speakers) are telephone recorded and $7k$ ($3k$ male and $4k$ female speakers) are microphone recorded.

4.2.2 Data processing, feature extraction and UBM creation

In the NIST 2012 SRE dataset, the microphone recorded interview data is available in the form of two-channel recordings. The primary channel (channel of interest) contains both the interviewee's (target speaker) and the interviewer's voice in similar amplitudes while the other channel only the interviewer's voice is dominant in terms of amplitude. The relatively lower amplitude interviewee's voice present in the second channel is masked with synthetic additive noise. To remove the interviewer's voice from the primary channel, a two stage process is followed. First, an adaptive two threshold energy based VAD, which is presented in 2.1 is employed to mark the interviewer's voice in the second channel. Then, using these markings, the interviewer's voice is removed from the primary channel. All other signal processing techniques and the acoustic feature

extraction method employed for the systems presented in this chapter remain same as that of the systems discussed in Chapter 3. All SV systems are developed in gender-dependent mode. Two gender-dependent UBMs of 1024 Gaussian mixtures are created and are kept common to all the developed systems. The UBM for modeling the male speakers is created using about 40 hours of telephone recorded speech data from 725 speakers. The UBM for the female speakers is created using about 50 hours of telephone recorded speech data from 1099 speakers.

4.2.3 Configuration of systems, parameter tuning and testing

For the purpose of tuning the system parameters and for the calibration of scores, we have developed an initial version for each of the systems under study and these systems are referred to as development (*dev*) systems. For this purpose, the NIST 2012 SRE training dataset is split into development train (*dev-train*) and development test (*dev-test*) sets. The *dev-test* set contains about 4000 speech segments of 30-100 seconds durations. A set of development trials (*dev-trials*) is created comprising approximately 80,000 claims keeping a ratio of 1:20 between true and false trials. The *dev* systems are trained on the *dev-train* dataset and evaluated on the *dev-test* dataset using these trials. The parameters of the different systems are tuned based on the performance for the *dev-trials*. The final version of all the systems are trained on the 2012 SRE training data with the tuned parameter values and evaluated on the 2012 SRE test data.

The i-vectors are derived using a *channel-conditioned* T-matrix having 800 columns by following the method described in [88]. The first 500 columns of the T-matrix are learned using telephone data. The remaining 300 columns are trained over the residues obtained by projecting the microphone data onto the columns learned using the telephone data. The exemplar dictionary for the XD-SR systems is created by pooling the supervectors representing the telephone training data for all the target speakers. Both the LD-SR and the DLD-SR systems use dictionaries having 1000 atoms trained by pooling the supervectors derived from telephone and microphone development data. The centered GMM supervectors for the JLD-SR and JDLD-SR systems are computed by normalizing the UBM-mean centered 1^{st} order statistics with the 0^{th} order statistics of the data computed with respect to the UBM. In the cases of JLD-SR and JDLD-SR systems, the speaker dictionaries of 800 atoms are trained by pooling the supervectors derived from telephone and microphone development data. The sparse representation vectors for the training and test data are estimated using the OMP algorithm with a sparsity constraint of 100.

For session/channel compensation, the i-vector CDS system uses an LDA projection of 400 dimensions and a full-rank WCCN projection. The Gaussian PLDA based i-vector system uses 500 dimensions to model the speaker subspace. For the XD-SR, LD-SR and the DLD-SR systems, the supervectors are normalized using pre-JFA. The implementation of the JFA module closely follows the method described in [16]. First, an eigen-voice matrix of 300 columns is trained on the telephone data from the development dataset. Then an eigen-channel matrix of 100 columns is trained on telephone data after removing the speaker components from it. It is then followed by learning another eigen-channel matrix of 100 columns on the the speaker and telephone channel components removed microphone data. Given data statistics, the compensated supervector is synthesized by multiplying the corresponding speaker factors estimated using the JFA with the eigen-speaker matrix. As shown in Equation 4.2, the channel dictionary in both the JLD-SR and the JDLD-SR systems has two partitions. These dictionary partitions contain 200 atoms each and are learned separately using telephone and microphone data.

The approaches followed for handling the multiple training examples of speakers and the cross channel trials are same as those in our submission to NIST 2012 SRE which is reported in [89]. The telephone and the microphone speaker models for each of the speakers are created by computing the sample-mean of the representations (i-vectors/sparse-vectors) of the telephone and the microphone training utterances available for that speaker, respectively. All test segments are scored against the matching channel models of the claimed speaker. As for some speakers the microphone model is not available, the telephone models are used in trials involving such speakers irrespective of the channel of the test data. Mapping of the scores to log-likelihood ratios, score calibration and fusion of systems are performed with logistic regression using the BOSARIS toolkit [61]. The *dev-trials* scores are used to train the parameters for the calibration and fusion.

4.2.4 Performance measures

For performance evaluation in this work, we have used a variant of detection cost which is the primary performance measure for NIST 2012 SRE tasks [87]. It is defined as the average of the normalized detection costs (C_{norm}) computed at two pre-defined operating points. Given the miss probability $P_{Miss|Tar}$, the known false-alarm probability $P_{FA|Known\text{-}Nontar}$ and the unknown false-alarm probability $P_{FA|Unknown\text{-}Nontar}$ computed by applying the corresponding threshold values on

the scores, the C_{norm} is defined as,

$$C_{norm} = P_{Miss|Tar} + \beta\{P_{Known} \times P_{FA|Known\text{-}Nontar} + (1 - P_{Known}) \times P_{FA|Unknown\text{-}Nontar}\} \quad (4.5)$$

where $\beta = \frac{C_{FA}}{C_{Miss}} \left(\frac{1-P_{Tar}}{P_{Tar}}\right)$. The constants C_{FA} and C_{Miss} denote the costs associated with false alarms and misses, respectively and both are set to 1. P_{Tar} is the prior probability of the target trial and P_{Known} is the prior probability that the non-target trial is from one of the evaluation target speakers. The value of $\log(\beta)$ is used as the threshold for computing the miss and the false-alarm probabilities. Let $C_{norm\text{-}1}$ and $C_{norm\text{-}2}$ denote the normalized detection costs computed with P_{Tar} set to 0.01 and 0.001, respectively while keeping the value of P_{Known} equal to 0.5 in both cases. The overall detection cost, $C_{DET\text{-}12}$, as it is defined for the NIST 2012 SRE, is then computed as,

$$C_{DET\text{-}12} = (C_{norm\text{-}1} + C_{norm\text{-}2})/2 \quad (4.6)$$

With these choice of parameters, the primary performance measure $C_{DET\text{-}12}$ evaluates the performance of the system at low false alarm operating points, i.e., it gauges the suitability of the system for high security applications. For assessing the non-application specific performances, EER is used.

4.3 Experimental results

In this section we present the performances of various proposed SR based SV systems developed and evaluated on the NIST 2012 SRE dataset. As given in Table 4.2, the evaluation data comprises of five different sensor/noise conditions. Table 4.3 shows the condition-specific performances along with their average (for the ease of comparison) for various proposed and contrast systems in terms of the chosen detection cost function ($C_{DET\text{-}12}$). The corresponding EERs averaged over the five test conditions are also listed in the table.

In case of LD-SR and DLD-SR systems, the supervectors are processed with the pre-JFA for channel compensation prior to sparse coding. On comparing with the contrast systems, the LD-SR system is found to result in 0.029 lower $C_{DET\text{-}12}$ than that of the i-vector CDS system and 0.054 higher $C_{DET\text{-}12}$ when compared to the i-vector PLDA system. On the other hand, the LD-SR system is found have 0.051 lower $C_{DET\text{-}12}$ than that of the XD-SR system, which supports our earlier arguments.

Table 4.3: The performances of the proposed SV system based sparse representation over simple and discriminative dictionaries along with using explicit (pre-JFA) and built-in session/channel compensation on the NIST 2012 SRE test dataset. The table also gives the performances of different contrast systems developed based on the existing SV approaches. Note that significant improvement is achieved over pre-JFA with proposed joint sparse coding approach for both simple and discriminative variants of the dictionaries.

	SV system name (key features)	Detection cost ($C_{DET\text{-}12}$)						% EER
		TC-1	TC-2	TC-3	TC-4	TC-5	**Avg.**	**Avg.**
Contrast	i-vector CDS (LDA & WCCN, CDS)	0.580	0.537	0.590	0.655	0.576	**0.588**	4.40
	i-vector PLDA (Gaussian PLDA, Bayes classifier)	0.470	0.599	0.458	0.470	0.530	**0.505**	5.25
	XD-SR (pre-JFA, Exemplar Dict., M-1 metric)	0.597	0.440	1.010	0.561	0.438	**0.610**	12.92
Proposed	LD-SR (pre-JFA, Learned Dict., CDS)	0.561	0.495	0.540	0.671	0.526	**0.559**	13.32
	DLD-SR (pre-JFA, Discrim. Learned Dict., CDS)	0.488	0.430	0.855	0.478	0.410	**0.532**	14.25
	JLD-SR (Joint Learned Dict., CDS)	0.527	0.378	0.561	0.421	0.366	**0.451**	8.98
	JDLD-SR (Joint Discrim. Learned Dict., CDS)	0.456	0.348	0.582	0.415	0.360	**0.432**	9.95

With a discriminative dictionary being used in the DLD-SR system, an additional reduction of 0.027 in $C_{DET\text{-}12}$ is achieved compared to the LD-SR system. Here we wish to highlight the fact that a similar extent of improvements are reported for the i-vector based systems also with the use of discriminative training [35, 90].

4.3.1 Joint sparse coding over speaker-channel learned dictionaries

The joint sparse coding based (JLD-SR and JDLD-SR) systems are found to result in a reduction of 0.1 in $C_{DET\text{-}12}$ when compared to the corresponding systems using pre-JFA for session/channel compensation. The observed improvement supports the arguments made while highlighting the shortcomings in the use of pre-JFA for the SR over learned dictionary based approaches in Section 4.1. To substantiate those, we have also analyzed the *mutual coherence* of the dictionaries in two cases. For the pre-JFA and the joint dictionary cases, the mutual coherence of the dictionaries are found to be 0.98 and 0.78, respectively. Further to assess the loss in speaker information with channel compensation in these approaches, an experiment was conducted on the discarded session/channel component of the supervectors. The $C_{DET\text{-}12}$ for the pre-JFA and the joint dictionary cases are found to be 0.985 and 1.156, respectively. Both these analytical results supports our arguments as well as justify the noted improvement. Finally on comparing the JDLD-SR system with the i-vector PLDA system a 14% relative improvement in $C_{DET\text{-}12}$ is noted.

From Table 4.3, it can be noted that despite the improved performance in terms of the $C_{DET\text{-}12}$ in comparison to the i-vector based systems, all SR based SV systems have resulted in poor EERs. This behavior is counter-intuitive and the reason for the same is investigated in the following section through the analysis of the distribution of scores.

4.4 Discussions on system characteristics

In this section we first analyze the score distributions of various systems to understand the behavior of their performance. It is followed by the discussion about, the robustness of the proposed systems in very low data and noisy conditions. The computational complexity involved in the proposed and the contrast systems are also compared.

4.4.1 Analysis of the distribution of scores

Among the two metrics considered for evaluating the performance of the various systems, the $C_{DET\text{-}12}$ is an application-dependent metric while the EER happens to be an application-

independent one. Despite the uniform costs assigned to the miss and false-alarm probabilities in computing $C_{DET\text{-}12}$, the two prior probabilities assigned to the target are kept very low. Further, the thresholds used to compute the miss and false-alarm probabilities are also derived using these low prior probabilities of the target. As a result, the operating points where the $C_{DET\text{-}12}$ is computed lie in the low false-alarm region unlike the EER.

Figure 4.2 shows the distributions of scores for the two best performing systems each from both the proposed (JLD-SR and JDLD-SR) and the contrast (i-vector CDS and i-vector PLDA) systems. The analysis is done for the evaluation condition TC-2 (clean telephone data) as it contains the largest number of trials among all the five evaluation conditions. On comparing Figure 4.2 (a) and Figure 4.2 (b), we note that the distributions of both the known and unknown false trial scores for the i-vector CDS system are Gaussian-like in nature while those are highly concentrated in the case of JLD-SR system. As the speaker representations are sparse in case of the SR based systems, with the cosine distance scoring, most of the false trial scores turns out to be zero (appear shifted from zero in the histograms for the scores being calibrated). Unlike this, for the i-vector based systems the representations are non-sparse and therefore are less likely to produce the false trial scores that are highly concentrated. The true trial score distributions for both the i-vector CDS and JLD-SR systems are found to have similar nature except for the fact that a number of trials in the JLD-SR systems produce very low scores on account of the sparsity in representation. Distributions of similar nature are observed for the i-vector PLDA and the JDLD-SR systems as shown in Figure 4.2 (c) and Figure 4.2 (d), respectively. On comparing the score distributions of the i-vector based and the proposed SR based systems, it can be noted that the latter exhibit much higher confidence in rejecting the false trials which leads to the lower $C_{DET\text{-}12}$ obtained. On the other hand, the scores of a few true trials in the SR based systems are found to be indistinguishable from those of the false trials which happen to be concentrated at zero value. To emphasize this fact, we further analyzed the true trials those produced very low score in the JDLD-SR system and the comparison of the histograms for the JDLD-SR and the i-vector PLDA systems are shown in Figure 4.3. It is to note that with respect to the two thresholds used in the decision making, the noted concentration of scores for JDLD-SR does not affect the $C_{DET\text{-}12}$ performance but alters the linearity of the change in miss probability. This explains why the EERs have turned out to be degraded in case of the SR based systems.

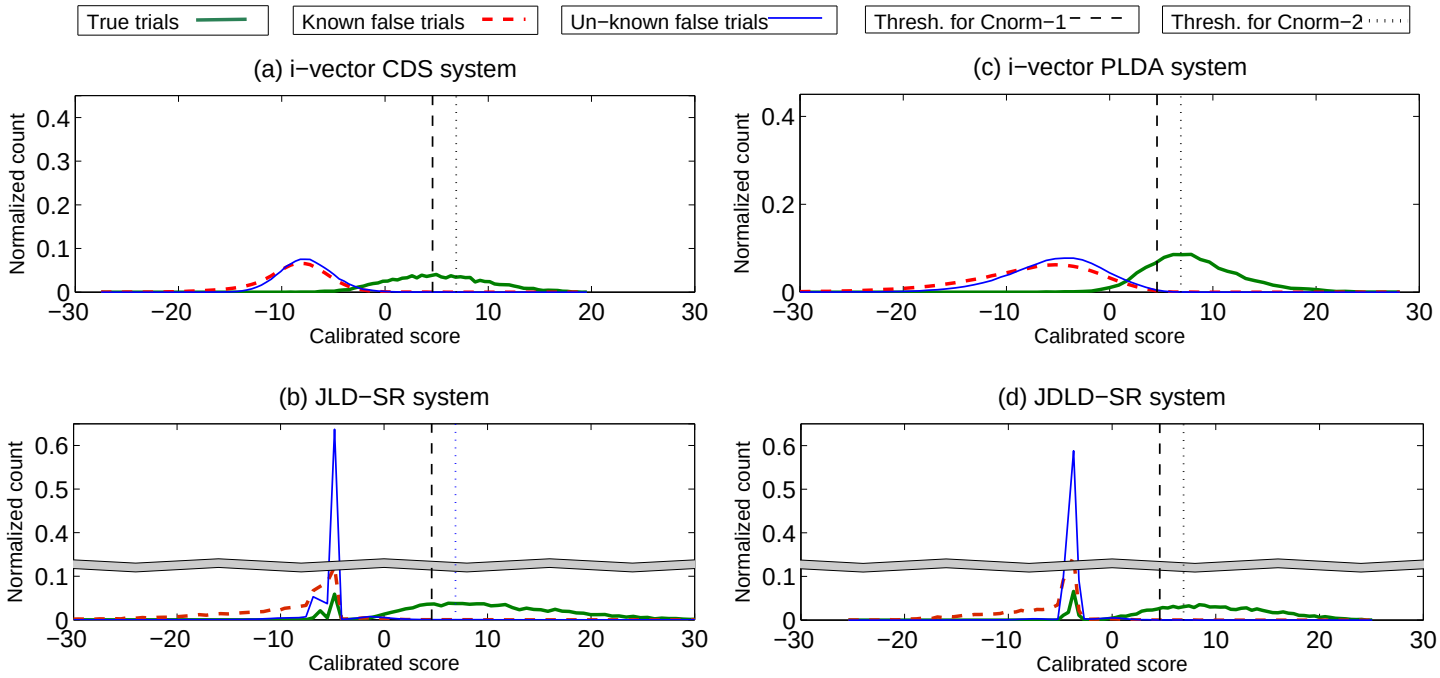


Figure 4.2: Histograms of the true and false trial scores for the proposed systems (JLD-SR and JDLD-SR) and the i-vector based contrast systems in the case of the evaluation condition TC-2.

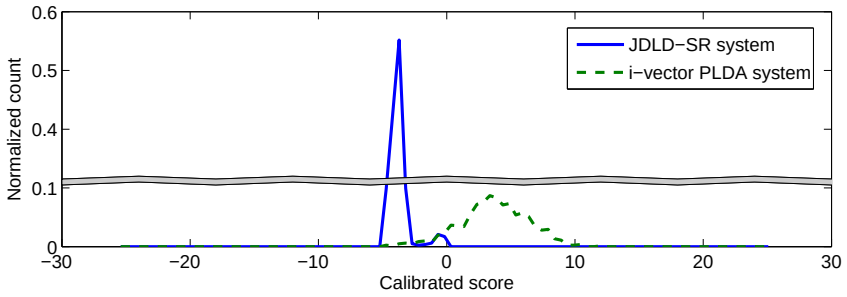


Figure 4.3: Histograms of a part of true trial scores in case of JDLD-SR and i-vector PLDA systems for the evaluation condition TC-2. These true trials are those which produced very low scores in JDLD-SR system. It highlights the fact that due to sparsity in representation in JDLD-SR system the scores of some of the true trials turn out to be very low unlike that in case of non-sparse i-vector PLDA system.

From the analysis of the histograms of scores, it appears that the sparsity of the speaker representations used in the SR based systems is the prime factor in achieving the improved low false-alarm performances at the cost of a degraded EER. To explain this trade-off further, we have evaluated the performances of the JDLD-SR system for sparsity values of 50 and 400 apart from the value of 100 chosen in this work. Figure 4.4 shows the DET curves for the JDLD-SR system with different selections of sparsity values along with that for the i-vector PLDA system, for the evaluation condition TC-2. It is to note that due to the failure of a few true trials, the DET curves for the JDLD-SR systems deviate considerably from linearity in comparison to that of the i-vector PLDA system, except in the low false-alarm region. On varying the sparsity from 100 to 400, a considerable reduction in the number of the true trial scores being exactly zero is expected. This fact is evident from the improved linearity of the corresponding DET curves. With the relaxation of sparsity from 100 to 400, the EER is noted to improve to 6.84 % from 9.95 % but at the same time the $C_{\text{DET-12}}$ is noted to degrade from 0.43 to 0.42. On changing the sparsity from 100 to 50, the EER is noted to degrade from 9.95% to 11.52% with a very small improvement in $C_{\text{DET-12}}$. These observations support our earlier reasoning.

It is to highlight that unlike the other NIST SREs, in the 2012 SRE the majority of the training data is derived from the older evaluation datasets. As we have used 2006-2010 SRE datasets for the system development, there is a possible overlap between the development and the training data. Note that both the proposed and the contrast systems happen to avail any possible advantage with this condition. Further, the 2012 SRE task includes false trials involving unknown (non-target) speakers and these allow us to assess the behavior of the speakers those are unseen to both training

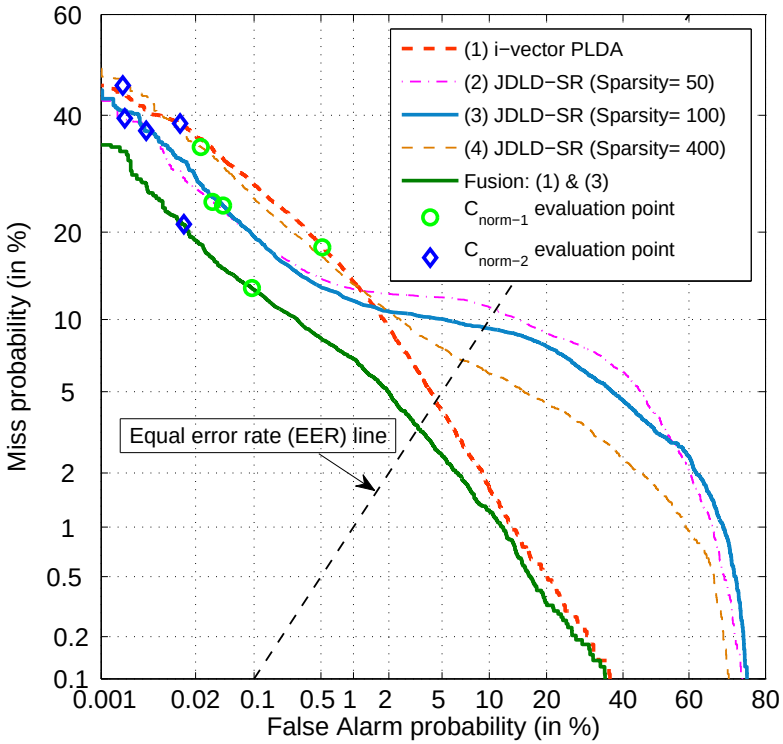


Figure 4.4: DET plots for the i-vector PLDA system, the proposed JDLD-SR system with varying sparsity and their fusion in the case of the evaluation condition TC-2. Note that with relaxation in the chosen sparsity value a significant improvement in EER for the JDLD-SR system is possible.

and development datasets. From Figure 4.2, we note that the edge of the SR based systems over the i-vector based systems is maintained for both the known and unknown false trials. In addition to that, the effectiveness of the proposed SR based system is already demonstrated in the work on NIST 2003 SRE data set with disjoint training and development datasets presented in Chapter 3.

4.4.2 Fusion of systems

With an appropriate choice of sparsity in representation, the SR based systems are able to achieve better low false-alarm rate performances than the i-vector based systems, but this improvement is achieved at the cost of a degraded EER. This motivated us to explore the fusion of the JDLD-SR and the i-vector PLDA systems. The fusion is performed using logistic regression with the help of the BOSARIS toolkit [61]. The scale and shift parameters for the fusion are optimized using the scores of the *dev-trials*. The performance for the fused system for all the five evaluation conditions and their average are given in Table 4.4. On comparing the averaged performances of the fused system with that of the component systems given, relative improvements of

Table 4.4: Performances of the fusion of the i-vector PLDA and JDLD-SR systems in various evaluation conditions. The averaged performances of the individual systems are also given for the ease of comparison. Note that in terms of both the measures significant improvements over the component systems are obtained with the fusion.

SV System name	Evaluation conditions	$C_{DET\text{-}12}$	% EER
Fusion of i-vector PLDA and JDLD-SR Systems	TC-1	0.375	6.24
	TC-2	0.286	3.35
	TC-3	0.592	5.79
	TC-4	0.291	3.79
	TC-5	0.239	3.81
	Avg.	**0.357**	**4.59**
i-vector PLDA	Avg.	0.505	5.25
JDLD-SR	Avg.	0.432	9.95

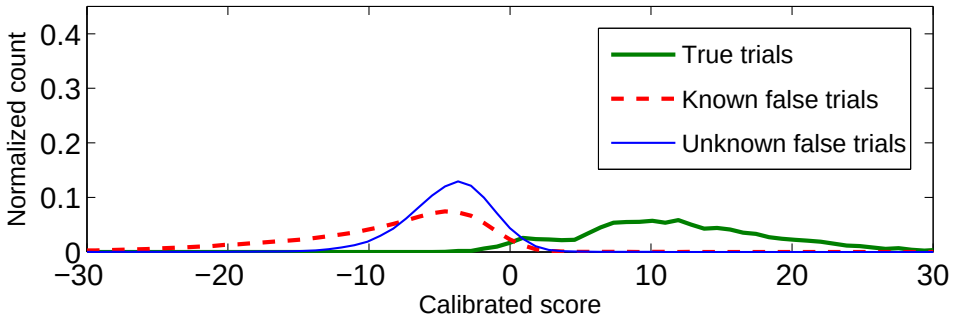


Figure 4.5: Histograms of the scores, for the evaluation condition TC-2, highlighting the increased confidence in scores with the fusion of the i-vector PLDA and JDLD-SR systems in comparison to those of the component ones.

17.4% in $C_{DET\text{-}12}$ and 12.6% in EER are noted over the corresponding best performances. For ease of comparison, the DET curve for the fused system for the TC-2 condition is also shown in Figure 4.4. It is interesting to note that with the fusion, the performance improvement is achieved for both low and high false-alarm operating regions of the system. Further, from the score-distribution of the fused system shown in Figure 4.5, two distinct attributes are noted. The fused system does not show any concentration of scores like that observed for the JDLD-SR system, and the distribution of true and false trial scores are now better separated than those of the component systems. Thus the fused system exhibits an enhanced performance exploiting the *complementary nature* of the two component systems.

4.4.3 Effect of low duration test data

The SR based approaches has resulted in an improved performance compared to that of the i-vector approach. As the i-vector representation is reported to degrade with decrease in amount of test data [91, 92], it would be interesting to study how the same affects the SR based systems.

For assessing the effect of decrease in test data duration in the systems performance, the evaluation condition averaged performances of the systems are repartitioned according to the test data duration break-ups given in Table 4.2. The duration-wise performances of the two best performing systems each from the proposed and the contrast systems are given in Table 4.5. Note that, for the 30 seconds case all the SR based approaches have outperformed the i-vector based system significantly while for the 100 and 300 seconds cases all the systems (except the i-vector CDS) have resulted in competitive performances. With majority of the evaluation test data having 300 seconds duration, it is obvious that the overall performance improvement noted in case of the SR based approaches are contributed by the significantly enhanced performance under the low duration test data trials. The i-vector representations are conventionally derived as the MAP point estimate while the sparse representations in the proposed approaches are derived as the l_0 regularized least-squares estimate. Both of these estimation approaches are expected to degrade with the reduction in the observed data and so the possible reason behind the noted behavior of the SR based systems is not obvious. In this context, we made a study of the histograms of scores for different test data duration conditions with the hope of finding some clues. Figure 4.6 shows the histograms of scores for the TC-2 evaluation condition split based on the three test data durations, i. e., 30, 100 and 300 seconds. On comparing the histograms, we note that with decreasing data duration both the true and false trial score distributions shift towards each other in case of the i-vector based system. On the other hand for the SR based system, only the distribution of the true trials are observed to shift towards that of the false trials, while no significant shift is noted for the false trials distribution. As a result of this, on computing the miss and false-alarm probabilities with respect to the two thresholds (involved in finding $C_{\text{DET-12}}$) marked in the figures, note that both these probabilities increase in case of the i-vector based system while only the miss probability increases for the SR based system. The possible explanation for this behavior of the distribution of the false trials in case of the SR based system is contributed by the fact that it is less likely to have the same set of atoms being involved in representing two different speakers.

Table 4.5: Performances averaged over the five evaluation conditions for the proposed systems (JLD-SR and JDLD-SR) and the i-vector based contrast systems for three data durations present in the NIST 2012 SRE test dataset.

SV Systems ↓	$C_{DET\text{-}12}$ (Avg.)		
Test data duration (sec) →	30	100	300
I-vector CDS	0.85	0.51	0.43
I-vector PLDA	1.02	0.39	0.29
JLD-SR	0.62	0.39	0.35
JDLD-SR	0.60	0.37	0.32

Even with the reduction in the test data duration, this nature remains unchanged keeping the false trial scores almost unaffected. To verify the same, we have plotted the training and test sparse representation vectors of a pair of randomly selected true and false trials corresponding to various test data durations in Figure 4.7 and Figure 4.8, respectively. The decrease in test data duration leads to unadapted GMM mean components in the supervector representations. Such supervector fails in selecting the same dictionary atoms selected by the training supervectors in sparse coding. This in-turn leads to more false rejections and hence degradation in overall system performance. On the other hand, due to sparsity in representation the number of false acceptances are less likely to increase in such conditions. These observations explain why the SR based system is noted to give lesser degradation over the i-vector based system under low data conditions.

4.4.4 Effect of additive noise in test data

Among the five test conditions in the NIST 2012 SRE primary task, two (TC-3 and TC-4) involve trials with realistic noise added to the test data. The two types of noise added are the *heating, ventilation and air conditioner* (HVAC) noise and the *crowd* noise. In these test conditions, the type and the level (5 dB or 15 dB) of the added noise are chosen in a random manner. Further, the trials across the clean and the noisy test conditions are different and hence the noise-robustness of the developed systems can not be judged by simply comparing the corresponding performances given in Table 4.3. To analyze the same in a controlled manner, we have performed a study using the test data corresponding to the clean phone channel trials, i.e., the TC-2 evaluation condition. The chosen data is added with HVAC and crowd noise separately at SNR values varying

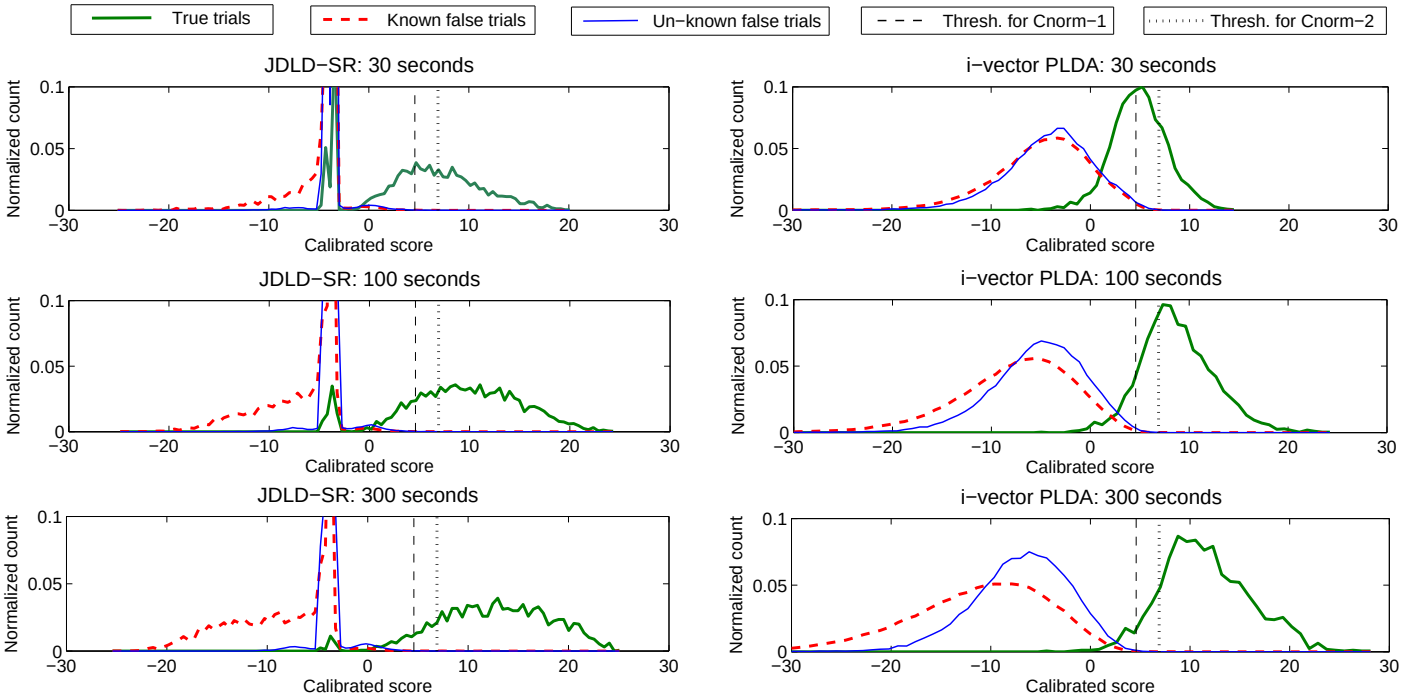


Figure 4.6: Histograms of the true and false trial scores for the proposed JDLD-SR and the i-vector PLDA based systems for various test data duration sub-conditions of the evaluation condition TC-2. Note that almost no change (w.r.t. to marked decision thresholds) is noted in the distributions of the false scores for the JDLD-SR case with reducing data duration unlike that for the i-vector case.

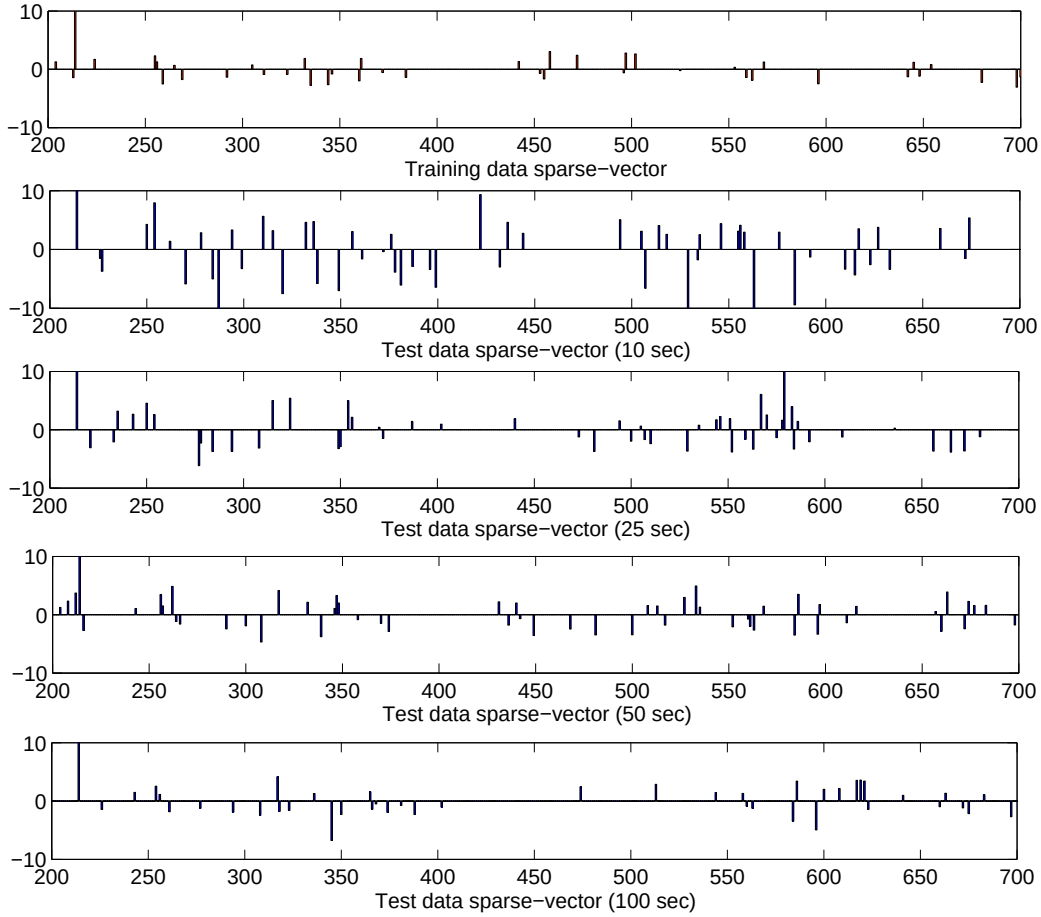


Figure 4.7: *Sparse representation vectors corresponding to training data and test data of different durations in case of a true trial.*

from 0-20 dB in steps of 5 dB. Ten examples each of the HVAC noise [1] and the *simulated* crowd noise (generated by adding speech data of a few hundred speakers derived from NIST 2005 SRE dataset) are used for this purpose. Figure 4.9 shows the performances of the JDLD-SR and the i-vector PLDA system for varying SNR values for the HVAC and the crowd noise cases. It is to note that for both type of noises, the relative improvement in terms of $C_{\text{DET-12}}$ of JDLD-SR system over the i-vector PLDA system is consistent for all SNR levels. From these results we infer that the SR based systems are as robust as the i-vector PLDA system to additive practical noise.

4.4.5 Computational complexity

From the implementation point of view, the proposed JDLD-SR and the i-vector PLDA systems differ in the computation of speaker representation and in the classifier stages only. Given the

[1] Downloaded from the *freesounds.org*

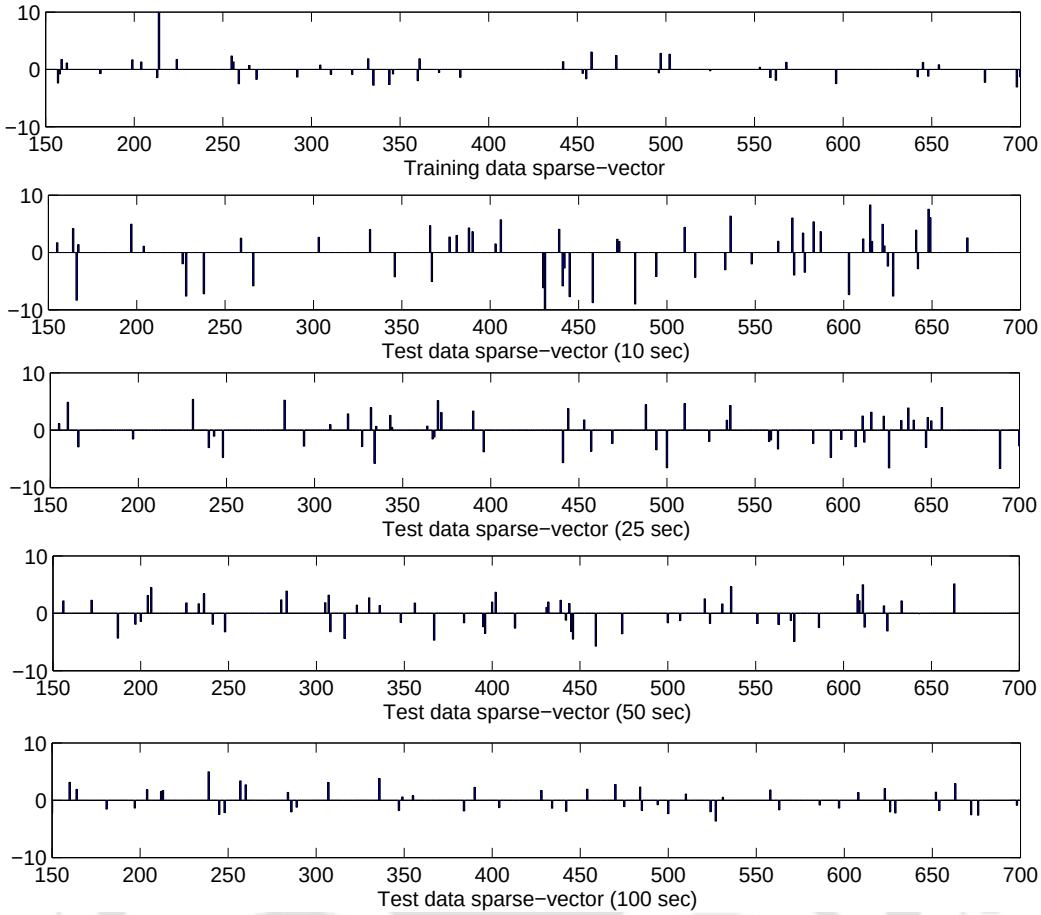


Figure 4.8: *Sparse representation vectors corresponding to training data and test data of different durations in case of a false trial.*

data statistics for a test utterance, the complexity involved in computation of verification scores for these two approaches are given in Table 4.6. For the comparison purpose, the total number of multiplications involved (for the parameter values chosen in this work) and the observed run-time for these systems are also given. We note that the proposed JDLD-SR system has a much lower complexity than the i-vector PLDA system. The major portion of the computational burden of the i-vector PLDA system is contributed by the i-vector computation step. It is to note that we have followed the default i-vector computation method [17] in this work, but there do exist a number of simplified i-vector computation methods [34–36]. Using those methods, about a 40 fold reduction in complexity of the i-vector computation over the default one is possible for the choice of parameters in this work. Thus the proposed JDLD-SR system can be considered to have a complexity comparable to that of the fastest i-vector PLDA systems reported.

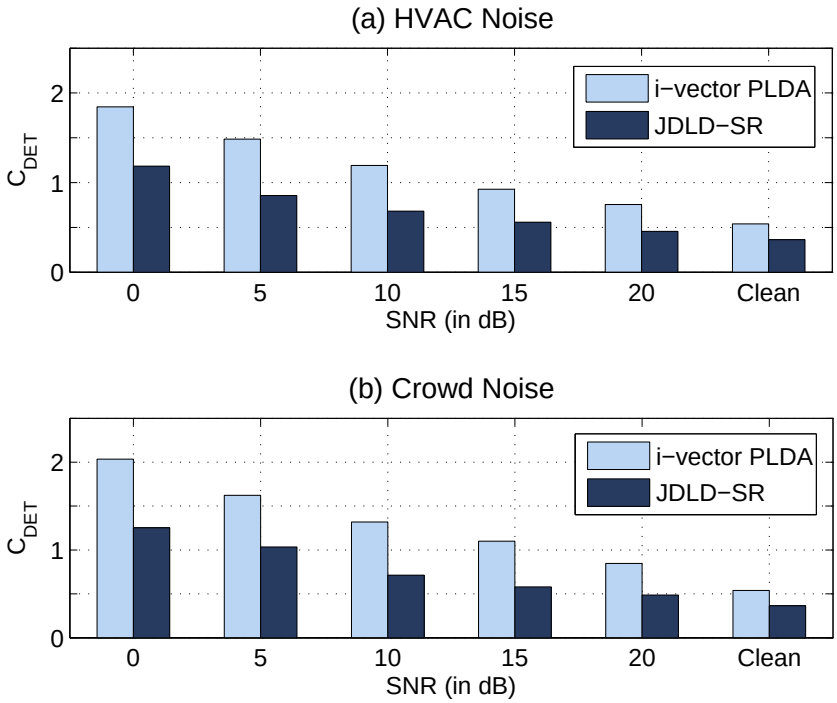


Figure 4.9: Bar-plots showing the performance of the proposed JDLD-SR and the i-vector PLDA based systems with noise added to the test data at varying SNR levels for the evaluation condition TC-2 for (a) HVAC noise and (b) crowd noise.

Table 4.6: Comparison of time complexity in the i-vector PLDA and JDLD-SR SV systems in computing the decision scores from data statistics (MFCC feature dimension $m = 39$, GMM size $p = 1024$, i-vector size $q = 800$, PLDA speaker space dimension $r = 500$, joint sparse representation vector size $s = 1200$, speaker sparse representation vector size $t = 800$, chosen sparsity constraint $l = 100$).

SV system	Multiplication complexity		Total mult. count	Runtime (ms)†
	Representation	Classifier		
i-vector PLDA	$\mathcal{O}(mpq + pq^2 + q^3)$	$\mathcal{O}(r^2)$	1.2×10^9	4,427
JDLD-SR	$\mathcal{O}(mps + l^2 s + ls + l^3 + t^2)$	$\mathcal{O}(t)$	6.2×10^7	97

† Computed using 64-bit MATLAB (R2010b) running on Intel Xeon 2.4 GHz 6-core CPU with 32 GB RAM in single thread default precision mode.

4.5 Summary

In this chapter, a joint sparse coding over composite speaker-channel dictionary approach which provides inherent session/channel compensation is proposed. When compared to an i-vector PLDA based contrast system on the NIST 2012 SRE primary task, the proposed approach provides 14% relative improvement in detection cost but with a degradation in EER. To exploit the complementary nature of the proposed and the contrast systems, a logistic regression based fusion

is employed. The resulted system is found to achieve relative improvements of 17.4% in detection cost and 12.6% in EER over the corresponding best performances of the component systems. On analyzing further, the proposed system is found to be more robust compared to the i-vector PLDA based system in the short-duration test data condition. On assessing the effect of additive noise, both the proposed and the contrast systems are found to exhibit a similar robustness. The computational complexity of the proposed systems is noted to be quite low when compared to the contrast system using the default i-vector extraction and is shown to be comparable even to the fastest i-vector extraction methods reported in literature. In the next chapter, we present our studies done on the use of data-independent projections for reducing the complexity of existing SV approach involving multiple data-dependent projections.

5

Low-complexity Data-independent Projections of GMM Supervectors

Contents

The current SV approaches predominantly involve very high-dimensional GMM mean supervectors as the representations of speaker utterances. As mentioned earlier, these representations are formed by concatenating the mean parameters of a GMM-UBM adapted with the speaker-specific data. The high-dimensional GMM mean supervectors are reported to have large redundancy and to reduce the same commonly the factor analysis based i-vector approach is used in the front-end of SV System. In that approach, a data-dependent low-rank projection matrix (T-matrix) is used to derive the low-dimensional i-vector representations [17] from supervectors. Often the i-vectors are further processed with discriminative linear transforms such as LDA and WCCN in the front-end to reduce the variabilities due to session and/or channel mismatch(es). State-of-the-art speaker verification systems use a Bayesian classifier with PLDA [60] or its variants [18, 19] for generating the final verification score.

Though the factor analysis based i-vector representation is quite effective in obtaining a compact and fixed dimensional representation of the speaker utterances, there are a few drawbacks in this approach. The algorithm used for the derivation of i-vectors is computationally complex and requires a large amount of memory to store the algorithm specific variables. In addition to that for the proper learning of the total variability matrix, a large amount of data is needed. There are already some works reported in literature for simplifying the i-vector computation. The approach proposed in [34] uses two approximations to reduce the number of computations. The first one is the use of a constant GMM component alignment across utterances given by the GMM-UBM weights. The second assumption is that the i-vector extractor matrix can be linearly transformed such that its per-Gaussian components are orthogonal. In [35] the factor analysis (FA) is performed on the pre-normalized centered GMM first-order statistics supervector to ensure that the statistics sub-vector of each of the Gaussian components is treated equally in the FA, which reduces the computational cost significantly. In addition, the matrix inversion term is simplified using a look-up table based approach which resulted in further speed up with only a small quantization error. A fast i-vector computation using the factored sub-space approach is also proposed that achieves a five fold reduction in the memory required for storing the T-matrix, without degradation in performance [36, 93].

In literature, data-independent projection approaches using random matrices are reported to provide a viable alternative to the data-dependent ones like PCA for reducing the dimensionality of very high dimensional vectors [37]. Motivated by that, we have first explored a few data-

independent projections to reduce the dimensionality of GMM mean supervectors in context of a sparse representation classification based speaker identification system. Among the various projections explored, the sparse random matrix based one demands very little computational resources only and resulted in a performance competitive to that of the i-vector based approach. It is followed by the exploration of such approaches for the SV task which is of more interest to the community. The major contributions reported in this chapter are: (1) The use of low-complexity sparse random projections for reducing the dimensionality of supervectors in context of the PLDA based SV system developed on a large multi-variability (NIST 2012 SRE) dataset, (2) A non-random data-independent projection with simple decimation of supervectors is proposed for dimensionality reduction and is shown to be as effective as the sparse random projection while having even lesser complexity, and (3) A multi-offset decimation diversity based SV system is proposed which outperforms not only the individual offset decimation based systems but also the default i-vector based system while still having lesser computational requirements.

The rest of this chapter is organized as follows. Section 5.1 describes the proposed idea of using data-independent projections for the dimensionality reduction of GMM mean supervectors. It also covers the various projection approaches explored in this work. The proposed idea has been initially explored in the context of an SRC based speaker identification task and the same has been presented in Section 5.2. Section 5.3 reports the exploration of low-complexity data-independent projections for the PLDA based SV task and the corresponding results are presented in Section 5.4. The proposed multi-offset decimation diversity based SV system is presented and the results are discussed in Section 5.5. The chapter is concluded in Section 5.6.

5.1 Data-independent random projections of GMM supervectors

Data-dependent projections like PCA or factor analysis, though are quite effective in reducing the dimensionality, require large amount of data and are complex to learn. One possible solution for this issue is the use of random projections which are considered as data friendly and low-complexity alternatives to the data-dependent projections. The basis of the use of random projection lies in the well-known Johnson-Lindenstrauss (JL) lemma [94]. It states that a set of n points in high dimensional Euclidean space can be mapped into an $\mathcal{O}(\ln n/\epsilon^2)$-dimensional Euclidean space such that the distance between any two points changes by only a factor of $(1 \pm \epsilon)$. Suppose we have an arbitrary matrix of data $\boldsymbol{X} \in \mathbb{R}^{d \times n}$. Given any $\epsilon > 0$, there is a mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$, for any

$k \geq 12\frac{\ln n}{\epsilon^2}$, such that, for any two columns $\boldsymbol{x_1}, \boldsymbol{x_2} \in \boldsymbol{X}$, we have

$$(1-\epsilon)||f(\boldsymbol{x_1}) - f(\boldsymbol{x_2})||^2 \leq ||\boldsymbol{x_1} - \boldsymbol{x_2}||^2 \leq (1+\epsilon)||f(\boldsymbol{x_1}) - f(\boldsymbol{x_2})||^2 \tag{5.1}$$

The random projection matrices are typically generated using Gaussian distributed random numbers. Apart from this, a few non-Gaussian random projections are also proposed in literature for high dimensional data processing [95]. In the following we describe different data-independent projection approaches explored in this chapter.

5.1.1 Normal random projection

For reducing the dimensionality using random projections, the original d-dimensional vector is projected to a k-dimensional ($k < d$) subspace, using a random $k \times d$ matrix $\boldsymbol{R}$. Using matrix notation where $\boldsymbol{X}_{d \times n}$ is the original set of n d-dimensional vectors,

$$\hat{\boldsymbol{X}}_{\boldsymbol{k \times n}} = \boldsymbol{R}_{\boldsymbol{k \times d}}\ \boldsymbol{X}_{\boldsymbol{d \times n}} \tag{5.2}$$

is the projection of the data onto a lower k-dimensional subspace. The JL lemma states that if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved. In this case, each entry r_{ij} of the projection matrix is a independently and identically distributed $\mathcal{N}(0,1)$ variable. The generation of such a projection matrix is much simpler than the one obtained using PCA or factor analysis. Utilizing the spherical symmetry among the normal random vectors, Dasgupta *et al.* [94] showed that even though the random subspace is not strictly orthogonal, it closely satisfies the JL bound as the chances of being close to orthogonal is high.

5.1.2 Sparse random projections

The use of normal random matrix for deriving projections avoids the need of training data and the complex learning process. At the same time, the random matrix based approach does not provide any computational advantage in computing the projections. Also, this approach does not produce representations as compact as that in the data-dependent cases. As a result, the size of the projection matrix in case of the random projection approach would be much higher than that in case of the data-dependent ones. For example, in cases where the dimension d of the data is very large (say 40,000) and is required to be projected to a significantly reduced dimension k (say 1000), for applying Gaussian random projection, $d \times k$ Gaussian numbers need to be generated.

The generation and storage of such a large matrix is also non-trivial. To address these issues, Achlioptas [96] proposed random projection matrices in which the elements are relaxed to be ± 1 and are chosen by thresholding a uniform random variable.

5.1.2.1 Achlioptas' matrix

Achlioptas' random projection matrix $\boldsymbol{R}$ is defined as a $k \times d$ matrix with $\boldsymbol{R}(i,j) = r_{ij}$, where $\{r_{ij}\}$ are distributed as follows:

$$r_{i,j} = +\sqrt{3} \begin{cases} +1 & \text{with probability } 1/6, \\ 0 & \text{with probability } 2/3, \\ -1 & \text{with probability } 1/6 \end{cases} \tag{5.3}$$

Let the reduced dimension $k \geq k_0 = \frac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3} \ln n$ where ϵ is the representation error, $\beta > 0$ is a parameter which controls the probability of success of the projections and n is the total number of data points to be projected. In [96], it is shown that the transformation using the Achlioptas' matrix satisfies the JL bound at least with a probability of $1 - n^{-\beta}$.

With the elements of the projection matrix being $\{\pm 1, 0\}$, the matrix multiplication reduces to additions only except for the final normalization factor. The sparsity of a vector/matrix refers to the number of non zero elements in it. In the Achlioptas' projection matrix, two-third of the elements are set to zero and as a result it becomes quite sparse. On comparing with classical normal projection matrix this sparse projection matrix allows for significant complexity reduction in the projection of large dimensional vectors.

5.1.2.2 Li's matrix

The sparsity of the Achlioptas matrix is further enhanced by Li *et al.* [95] by introducing a sparsity control parameter s in Equation 5.3 as defined below,

$$r_{i,j} = +\sqrt{s} \begin{cases} +1 & \text{with probability } 1/2s, \\ 0 & \text{with probability } 1 - 1/s, \\ -1 & \text{with probability } 1/2s \end{cases} \tag{5.4}$$

It is to note that the Achlioptas matrix corresponds to $s = 3$ in the Li's formulation. For the value of $s = 200$ used in this work, the sparsity $(\frac{1}{s})$ of the Li's matrix turns out be 0.5%.

5.2 Application of data-independent projections for SRC based speaker identification

In this section the initial work done exploring the data-independent dimensionality reduction of GMM supervectors in the context of an SRC based speaker identification is presented. The projections obtained by data-independent dimensionality reduction of GMM supervectors are used as utterance representations for the speaker identification system employing sparse representation over exemplar dictionary. The creation of the exemplar dictionary is done following the procedure already described in Section 3.2.1. To assess the quality of dimensionality reduction of supervectors with data-independent projections, we have contrasted their performances with the SRC based speaker identification systems using the original GMM mean supervector and the i-vectors as speaker representation. A direct cosine distance scoring of GMM supervector based and an i-vector PLDA based speaker identification systems have also been developed and evaluated for the purpose of benchmarking the performance of proposed approaches. All the identification systems built follow the one-against-all scoring procedure.

5.2.1 Session/channel compensation

For the SRC based identification systems with original GMM supervectors as speaker representation, the session/channel compensation is performed using the simplified JFA approach as described in Section 2.3.1. In cases of the i-vector representation based SRC and CDS systems, the LDA and WCCN transforms are used for session/channel compensation. On the other hand, for the SRC based systems using the representations derived through the proposed data-independent projections of supervectors, the direct linear discriminant analysis (DLDA) [97] followed by WCCN is used for session/channel compensation. The use of DLDA is necessitated by the fact that the data-independent projections can not be as compact as the i-vectors. Note that, in the experimental setup used the number of training examples available are limited whereas the dimensionality of the supervectors is quite large. As per the JL lemma, the dimensionality of the projected vector is related logarithmically to the number of data points the dataset, thus the supervectors in our case can not be projected to a very compact size as the number of examples available are finite. In such conditions the computation of the LDA matrix become infeasible as the within-class covariance matrix of the training data becomes singular. This is referred to as the 'small sample size' problem in literature and to overcome the same we have used the DLDA [97] in case of the

Table 5.1: Details of the experimental data used for the study of the data-independent projections performed in context of SRC based speaker identification.

Dataset	Channel type	No. of segments		No. of speakers	
		Male	Female	Male	Female
Training	Telephone	1101	1648	138	209
Test	Telephone	981	1251	136	206
	Microphone	360	664	13	21
Development	Telephone	7327	12248	769	1160
	Microphone	3279	4030	335	423

data-independent projections.

5.2.2 Experimental setup

For evaluating the proposed approaches and to compare it with the contrast systems, the NIST 2005 SRE database has been used. The 2005 SRE database is primarily designed for speaker verification and contains multiple training and test conditions [98]. For the speaker identification experiments reported in this work, the training data is taken from *8-conversation 2-channel* condition and testing was done with both *1-conversation 2-channel* and *1-conversation aux-mic* test sets. The *1-conversation 2 channel* test set contains speech segments from telephone conversations and the *1 conversation aux-mic* test set contains speech segments recorded using an auxiliary microphone.

The implementation of systems based on data dependent approaches such as i-vector, JFA, LDA and WCCN require a large matching development dataset. Usually such development data for NIST SRE systems are derived from the previous evaluation datasets. From 2004 SRE onwards the evaluation data are derived from the Mixer corpus rather than from the Switchboard corpus [99]. As a result there is a lack of sufficient matching development data for 2005 SRE experiments and this issue is already reported in [16]. To overcome this we have derived data from 2006 SRE, 2008 SRE and 2010 SRE [99] datasets for the system development. The number of speakers and number of speech segments present in the training, test and development data sets are summarized in Table 5.1.

The basic signal processing and acoustic feature extraction methods employed is same as that of the systems discussed in the Chapter 3. Two gender-dependent GMM-UBMs of 1024 mixtures each are used in building all the speaker identification systems. An eigen-voice matrix of 400

Table 5.2: Performances of various speaker identification systems with session/channel compensation on the *8-conversation 2-channel* task in NIST 2005 SRE. In cases of random projection matrices the experiments are repeated for 20 times and the averaged performances are reported.

System	Projection Matrix	Projected Dim.	Recognition Rate (%)			
			Telephone		Aux. Mic	
			Male	Female	Male	Female
CDS classifier GMM supervectors			**87.66**	**83.73**	**71.20**	**56.83**
PLDA Bayesian classifier on i-vectors			**88.58**	**85.01**	**72.12**	**58.03**
SRC on GMM Mean Supervectors	No projection	39 k	**90.54**	**86.15**	**72.19**	**54.12**
	Total variability: i-vector (Data-dependent)	200	82.36	79.12	55.34	41.18
		400	86.34	82.29	65.93	47.93
		600	89.89	86.73	71.93	53.02
		800	**92.35**	**88.24**	**74.86**	**56.12**
		1000	90.02	86.14	63.01	54.41
	Random: Normal (Data-independent)	400	48.41	44.01	25.41	9.93
		1000	70.03	66.99	45.81	30.48
		2000	78.59	74.50	57.54	40.74
		4000	85.52	81.54	65.92	47.41
		8000	**88.99**	**84.82**	**71.22**	**52.15**
	Random: Achlioptas' (Data-independent)	400	45.05	41.05	22.62	12.19
		1000	67.17	63.99	49.16	31.03
		2000	79.51	75.15	59.77	41.57
		4000	85.62	81.90	68.15	50.42
		8000	**89.19**	**85.01**	**71.50**	**53.28**
	Random: Li's with $s = 200$ (Data-independent)	400	47.70	44.45	26.53	13.21
		1000	67.27	63.12	41.62	23.15
		2000	79.20	75.05	57.26	41.23
		4000	85.52	81.98	66.75	48.46
		8000	**89.09**	**85.81**	**71.78**	**53.34**

columns and an eigen-channel matrix of 200 columns are used in the JFA implementation. A channel-conditioned T-matrix of 800 columns is used to derived the i-vectors. The Gaussian PLDA system uses 600 dimensional speaker subspace. For performing the session/channel compensation in different SRC systems, 300 dimensional LDA and DLDA projections followed by WCCN with no reduction in dimension are applied for the i-vector and the data-independent projection cases, respectively.

5.2.3 Experiments and results

To study the effect of the dimensionality reduction of GMM mean supervectors in context of SRC, data-dependent (i-vector) and a number of data-independent projection methods are explored. The experiments are performed in a gender-dependent manner. As there is no auxiliary

microphone data in the training set, so all speaker models are trained using only telephone data. Each of the systems are tested with both telephone and auxiliary microphone test data. The performances of different systems developed are reported with the suitable and best performing session/channel compensation method(s) only. Table 5.2 shows the performance of the proposed as well as the contrast speaker identification systems in terms of % recognition rate.

For the telephone data case, the performances averaged over male and female for the SRC system with original supervector and the contrast systems are found to be quite close. For the SRC system with data-dependent dimensionality reduction, i-vector representations of different dimensions are derived and used. For data-independent dimensionality reduction case, first Gaussian (normal) random matrices with varying projection dimensions are explored. As described in previous section, for random projection the limit on the reduced dimension k to satisfy the JL-bound is given by $k \geq 12(\frac{\ln n}{\epsilon^2})$. In our case, the number of total training examples n is around 1000 and assuming an error of 10 % in representation ($\epsilon = 0.1$), the nominal value of the term $\frac{\ln n}{\epsilon^2}$ comes out to be 691, so $k \geq 8300$. From Table 5.2, it can be noted that the performances for all the three random projection approaches with $k = 8000$ turned out to be only slightly degraded compared to that for the no-projection case, for almost all test data conditions. These results are in coherence with the JL-bound. In contrast to the random projections, the data-dependent projections are learned from the data itself. As a result, the data-dependent projections are expected not only to be more compact but also more effective in removing the noisy dimensions than the random projections. This explains the slightly improved optimal performance observed in case of the i-vector compared to those in the random projection cases.

5.2.3.1 Effect of different realizations of projection matrices

In case of random projections, the classification performances could have some variation for different realizations of the projection matrix. To assess the extent of that variation in performances, all experiments in case of random projections are repeated for 20 times and the averaged performances are reported in Table 5.2. The standard deviation (SD) of the performances for different random projection matrices for the male-telephone test case are shown in Figure 5.1. It is to highlight that a similar behavior is also expected to be observed when the T-matrix for the i-vector approach is trained with different datasets. To explore that we actually require multiple sets of development data of similar size and nature to the one used in this work. As we do not have access

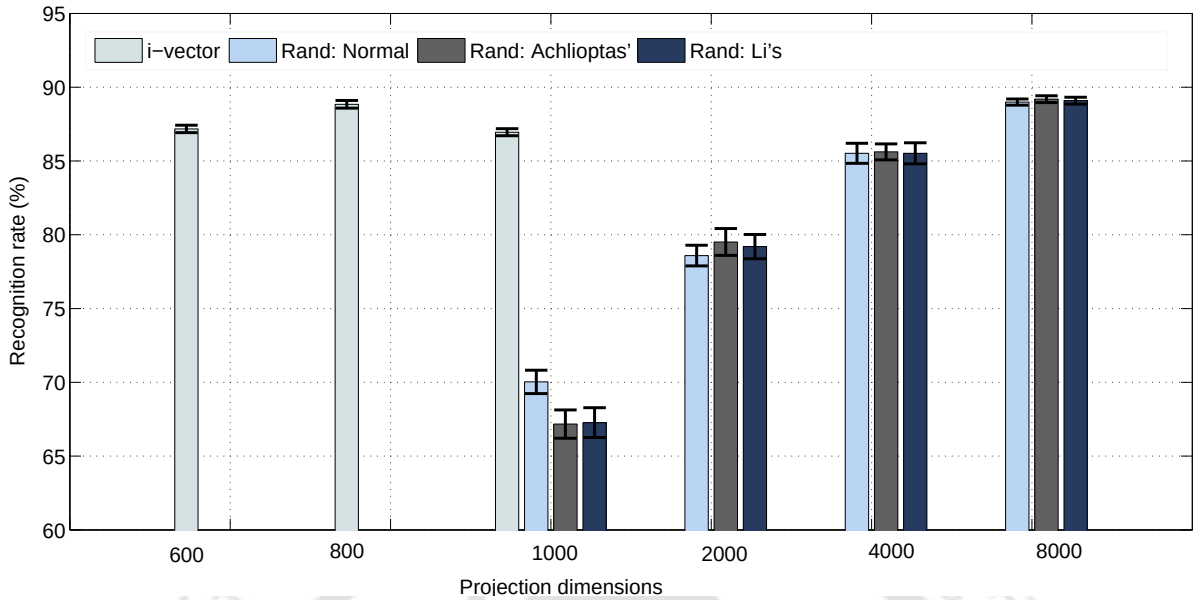


Figure 5.1: Error bars showing the deviation in the performances due to different realizations of the projection matrix in case of data-dependent and data-independent projection based systems, for the male-telephone test data. Note that, in the i-vector case the performances are lower than those given in Table 5.2 due to reduced development data being used.

to additional development data, we have created 10 random selections of half the original size from the available development dataset. Using these data subsets separate T-matrices are learned and the corresponding i-vector system performances are evaluated. The SD of those performances for the male-telephone test case are also shown in Figure 5.1. Needless to mention that these performances for the i-vector cases are much lower than those given in Table 5.2 due to the lesser data being used in the T-matrix training. From Figure 5.1, it is to note that the performances for both the i-vector and the random projection based approaches with *optimal choice of dimensions* exhibit similar extent of variation in the performance.

5.2.3.2 Random matrix: Tuning of sparsity

As mentioned earlier, the sparsity in case of Li's matrix can be controlled to reduce the computational complexity of the random projections of data unlike the total variability and the normal projection matrices. To get an understanding about the relative sparsity of different random projection matrices, the image plots of their absolute values are illustrated in Figure 5.2. In Table 5.2, the sparsity control parameter s for the Li's matrix is chosen to be $s = \sqrt{d} = 200$. This choice of the value of s results in the projection matrix of size 8000×39936 having only 0.5 % non-zero

Table 5.3: Performance of the SRC based speaker identification system using supervectors with reduced dimension of 8000 using Li's matrix for varying values of sparsity for the case of male-telephone test data. Note that in each case, the experiment is repeated for 20 times and the averaged performance along with the standard deviation (SD) is reported.

Parameter s	% Proj. matrix sparsity $(\frac{1}{s} \times 100)$	Avg. Recog. Rate (%)	SD (σ)
100	1.000	89.14	0.14
200	0.500	89.09	0.22
500	0.200	89.28	0.28
1000	0.100	89.15	0.34
2000	0.050	89.00	0.45
4000	0.025	88.77	0.48

elements. There is a scope for tuning the sparsity parameter and the same is performed for the male-telephone test case with reduced dimension of 8000. The corresponding performances are shown in Table 5.3. It is to note that even with a matrix having only 0.05 % non-zero elements, the performance is found to be very close to that of the best performance achieved for 10 fold lesser % sparsity. On further increasing the sparsity of the projection matrix, the performance starts degrading further.

(a) Normal projection matrix

(b) Achlioptas' projection matrix

(c) Li's projection matrix with $s = 200$

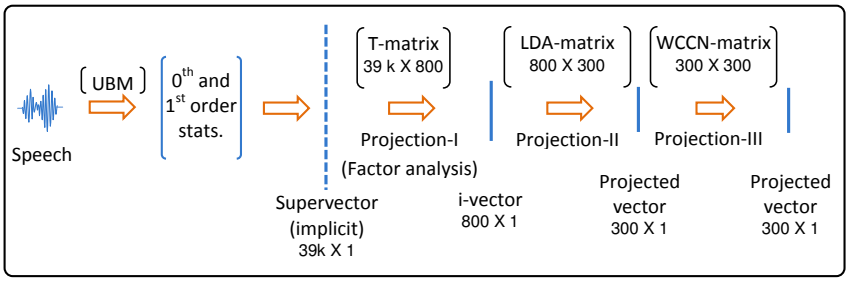
Figure 5.2: Image plots of the absolute value of different projection matrices of size 10×50 for illustrating their sparsity. Note that the Li's projection matrix is highly sparse and entries of the projection matrix are quantized to $\pm 1, 0$ unlike that in the normal projection matrix.

5.2.3.3 Complexity reduction with random projections

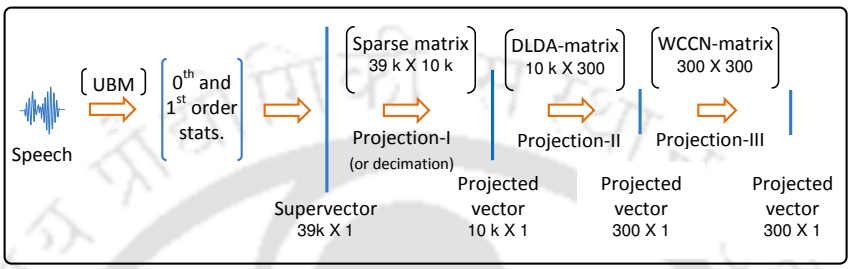
As given in [34], the multiplication complexity in computing an i-vector using Equation 2.12 is $\mathcal{O}(mpq+mq^2+q^3)$ where m is the size of GMM, p is the dimension of the MFCC feature and q is the size of the i-vector. The number of floating point multiplications required turns out to be 1.19×10^9 for the parameters used in this work. In case the Li's matrix as mentioned earlier, no multiplications are involved except for the final scaling in finding the projections. So the multiplication complexity for the Li's matrix case is the same as the projection dimension which is 8000 in the optimal case. The estimate of the complexity involved in the channel compensation (LDA/DLDA+WCCN) is $\mathcal{O}(st+t^2)$, where s and t denote the input and output vector dimensions respectively. In case of the i-vector based system, the computational complexity in channel compensation is 2.4×10^6 which is negligible compared to that of the i-vector computation and hence no significant increase in the overall computational complexity. On the other hand, in case of the Li's matrix based system, the computational complexity in channel compensation dominates that of the initial projection and the overall complexity turns out to be 2.4×10^6 for the chosen parameters. An additional attribute of the Li's matrix beyond the computational saving in projection is that it does not require a floating point representation. The T-matrix used for computation of i-vectors in this work occupies about 128 MB of storage space. In contrast, the Li's matrix being highly sparse, one needs to save only the indices of the nonzero elements along with the sign information. As a result, the Li's matrix with $s = 200$ occupies about 6 MB of storage space.

5.3 Data-independent projections of supervectors for PLDA based speaker verification

In this section we explore the low-complexity data-independent projections for the SV task which is more popular compared to the SI task considered in the previous section. A Gaussian PLDA based approach with length normalization of speaker representation vectors as described in Section 2.4.3 is used for building the SV system. As noted in previous section, all the data-independent projections have performed quite competitive while some of them are more advantageous in terms of computational complexity. So for the studies performed with the SV system, only Li's sparse projection matrix is considered for having the least complexity. In addition, we have also explored the decimation of GMM supervectors as a method of non-random data-independent dimensionality reduction. The details and advantages of the decimation process as a



(a) I-vector case



(b) Proposed low-complexity projections cases

Figure 5.3: Front-end projections involved in the PLDA based SV system using (a) the i-vector and (b) the proposed low-complexity projections as the speaker representations. In the i-vector case, the supervector is implicit and is shown for the sake of comparison. Note that the first projection in the proposed approach is data-independent whereas all the three projections are data-dependent in case of the i-vector approach.

candidate for dimensionality reduction is described in the following subsection. The studied low-complexity data-independent projections are contrasted with the data-dependent factor analysis based i-vector representations. In this study, i-vector representations computed using both the default approach [17] and a simplified (for better speed) approach [35] are considered. The various front-end processing steps involved in the i-vector and the proposed low-complexity projection based SV systems are shown in Figure 5.3. The illustrations highlight the kind and the size of different projections involved in both the approaches. In the i-vector case, the first projection is data-dependent whereas in the proposed case it is data-independent. The next two projections in both the cases are intended for the removal of session/channel variability from the representations. Those projections are required to be supervised and hence are data-dependent. For the i-vector based SV system, LDA followed by WCCN is used for session/channel compensation. In case of the data-independent projections, due to the 'small sample size' problem as discussed in Section 5.2.1, DLDA followed by WCCN is used for session/channel compensation.

5.3.1 Use of decimation as a low-rank projection

Decimation is the process of re-sampling the data at a lower rate. In this, every i^{th} sample in the data is retained and the shift i is referred to as the decimation factor. Decimation process can also be interpreted as a low-rank projection of the original vector as shown below with the help of an example (for no-offset case).

$$[x_1\ x_3\ x_5]^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} [x_1\ x_2\ x_3\ x_4\ x_5]^T \tag{5.5}$$

Based on the matrix interpretation of the decimation operation, we argue that although the projection matrix involved is deterministic, yet in effect it can closely approximate a realization of a highly sparse random projection matrix. The sparsity of the projection matrix in this case depends upon the chosen decimation factor. For a decimation factor of 4 used in this work, only 0.0025% of the elements in the projection matrix turn out to be non-zero. In actual practice, the decimation process is implemented using a simple selection operator, thus it avoids the need of both multiplication and addition in computing the projections and therefore it entails even lesser complexity.

5.3.2 Database and system parameters

Evaluation of the proposed approaches is done on the NIST 2012 SRE database [87] and the details of the same are already presented in Section 4.2.1. The experimental setup including the basic signal processing and feature extraction techniques employed are same as those used for the work reported in Chapter 4 unless specified otherwise. For deriving the speaker representations, the 0^{th} and 1^{st} order Baum-Welch statistics of the target speech data are computed with respect to the corresponding UBM. The size of the i-vector is chosen as 800 for both the default and simplified implementations. This choice is supported by an i-vector dimension tuning experiment reported in [100]. The default i-vectors are derived using the computed statistics and a channel-conditioned T-matrix having 600 telephone and 200 microphone columns as described in [88]. The simplified i-vectors are computed using the codes made available by its developer [101].

For the proposed approaches, the GMM mean supervectors are computed by normalizing the UBM mean centered 1^{st} order statistics with the 0^{th} order statistics and then projected to a low

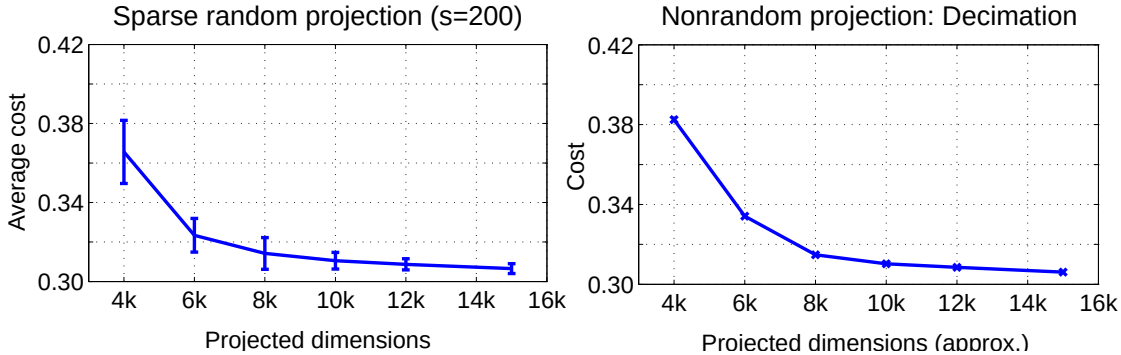


Figure 5.4: Tuning of the projection dimensions for the sparse random and decimation cases on the *dev-test* set. For the sparse random projection case, the averaged performance along with its standard deviation computed over 10 realizations are shown.

dimensional space. In case of the sparse random projection matrix, the optimal size can be determined based on the JL lemma as discussed earlier. In the chosen evaluation task (NIST 2012 SRE), there are 1931 speakers ($n = 1931$), assuming an error of 10% in projection representation ($\epsilon = 0.1$), the bound on the projection dimension ($k \geq 12\frac{\ln n}{\epsilon^2}$) turns out to be 9078. We have tuned the projection size around this theoretical value on *dev-test*. Figure 5.4 shows the mean performances along with standard deviation computed over 10 realizations for the sparse random projection based systems. Based on tuning, a projection dimension of 10 k is chosen for both the systems. The tuning of projection dimension for the decimation with zero-offset case is also shown in Figure 5.4 and 10 k (≈ 9984, which corresponds to a decimation factor of 4) turns out to be the suitable size in this case too. The channel-conditioned LDA/DLDA matrices with 300 projection dimensions and full-rank WCCN matrix are used in appropriate systems for session/channel compensation. The telephone and microphone speaker models for each speaker are created by taking the sample-mean of the representations of the telephone and microphone training utterances available for that speaker, respectively. All test segments are scored against matching channel models of the claimed speaker. As for some speakers the microphone models are not available, telephone models are used in trials involving such speakers irrespective of the channel of the test data. The normalized detection cost defined for the NIST 2012 SRE task ($C_{\text{DET-12}}$) presented in Section3.4.2 is used as the primary performance measure for the various systems developed. In addition, EER values are also computed and reported.

Table 5.4: Performances of the PLDA based SV system with different types of data-independent projections and that with the i-vector representations on the NIST SRE 2012 primary task, in terms of EER and normalized detection cost $C_{DET\text{-}12}$.

1^{st} projection matrix		Evaluation condition	EER (%)	$C_{DET\text{-}12}$
Data-dependent	T-matrix: i-vector (Default)	Interview	6.55	0.492
		Phone call	4.90	0.557
		Interview noisy	5.45	0.483
		Phone call noisy	4.54	0.512
		Phone call noisy env.	4.84	0.528
		Average	**5.26**	**0.514**
	T-matrix: Simplified i-vector [35]	Interview	7.38	0.568
		Phone call	5.98	0.641
		Interview noisy	6.37	0.519
		Phone call noisy	5.60	0.538
		Phone call noisy env.	5.67	0.581
		Average	**6.20**	**0.569**
Data-independent	Sparse random (with $s = 200$)	Interview	7.68	0.582
		Phone call	5.89	0.675
		Interview noisy	6.63	0.517
		Phone call noisy	5.66	0.557
		Phone call noisy env.	5.75	0.613
		Average	**6.32**	**0.588**
	Decimation (with no-offset)	Interview	7.68	0.584
		Phone call	5.82	0.664
		Interview noisy	6.65	0.523
		Phone call noisy	5.58	0.568
		Phone call noisy env.	5.73	0.596
		Average	**6.29**	**0.587**
No projection		Interview	7.19	0.534
		Phone call	5.82	0.614
		Interview noisy	6.25	0.502
		Phone call noisy	5.34	0.527
		Phone call noisy env.	5.53	0.526
		Average	**6.03**	**0.541**

5.4 Results and discussions

The performances of the PLDA based SV system using the proposed projection representations as well as that with the default and the simplified i-vector [35] representation are given in Table 5.4 for the five different evaluation conditions. The i-vector, being a data-dependent projection, provides not only dimensionality reduction but also an improvement in classification performance over the supervector representations through the de-emphasis of the noisy measurements. So for a

fairer comparison, the performances of the data-independent projections should be contrasted to that with no-projection, i.e., using the original supervectors. Table 5.4 also shows the performance of the SV system with the DLDA matrix being directly applied to the original supervectors. The proposed sparse random projection and the decimation based systems are found to give competitive performances. On comparing the best performing projection approach out of the proposed ones (i. e., the decimation case) with the default i-vector representation, a degradation of 0.073 in $C_{\text{DET-12}}$ and 1.03 in % EER is noted when averaged over all the five evaluation conditions. On the other hand, in comparison to the simplified i-vector, no significant degradation in performance is noted with the proposed projections. When compared to the the no-projection case, the decimation based system is noted to have a reduction of 0.046 in $C_{\text{DET-12}}$ and 0.26 in % EER. The main motivation behind the use of proposed projections lies in the reduction of computational complexity and the same is discussed in the following sub-section.

5.4.1 Computational complexity

The order of the multiplication complexity and the memory requirement involved in the first two projections in the proposed low-complexity projection based systems and those in the default as well as simplified i-vector based systems are given in Table 5.5. For the comparison of the actual complexity involved, the observed runtime of the different systems are also given. The computations and memory usage involved in the third projection (WCCN) and the PLDA-classifier are relatively small and are same for all the systems, so these are not considered in the comparison. As mentioned earlier, the projection with sparse random matrices can be realized with additions only, except for the final scaling. This makes the corresponding system much faster than both the default and simplified i-vector based systems. On the other hand, the decimation approach does not involve multiplication at all and its runtime turns out to be 40 times smaller to that of the considered simplified i-vector approach [35].

In addition to the reduced projection complexity, both the sparse random projection and the decimation based systems are also attractive in terms of the storage space required for the projection matrix in comparison to the i-vector based ones. In case of the sparse projection matrix, one needs to save only the indices of the nonzero elements along with the sign information which in turn do not require floating point representations. The decimation approach has the least storage requirement as it avoids the need of projection matrix completely.

Table 5.5: Comparison of the multiplication complexity and the memory usage in the first two front-end projections of different proposed and existing SV systems. (GMM size $m = 1024$, MFCC feature dimension $p = 39$, i-vector size $q = 800$, random projection size $r = 10k$, and LDA/DLDA projection size $s = 300$).

Projection-I approach	Multiplication complexity		Runtime (ms)†	Memory usage (in MB)			
	Proj.-I	Proj.-II		Fixed		Algo. specific	Total
				Proj.-I	Proj.-II		
i-vector (default)	$\mathcal{O}(mpq + mq^2 + q^3)$	$\mathcal{O}(qs)$	4,620	255	2	5250	5507
i-vector (simplified)	$\mathcal{O}(mpq)$	$\mathcal{O}(qs)$	273	255	2	51	308
Sparse random	$\mathcal{O}(r)$	$\mathcal{O}(rs)$	13	5	24	-	29
Decimation	-	$\mathcal{O}(rs)$	6	-	24	-	24

† Computed using 64-bit MATLAB (R2010b) running on Intel Xeon 2.4 GHz 6-core CPU with 32 GB RAM in single thread default precision mode.

In addition, the proposed approaches does not have any algorithm-specific memory requirement during the computation of representation, unlike the i-vector computations. Hence, the overall memory requirement for both the sparse random projection and the decimation based systems are noted to be about 10 times lesser than that for the simplified i-vector based system.

Recently, another fast i-vector computation using the factored sub-space approach is proposed that achieves a five fold reduction in the memory required for storing the T-matrix, without degradation in performance [93]. The complexity of this approach is slightly higher than the eigen-decomposition based approach reported in [34] which in turns has a complexity same as that of the simplified i-vector considered in this work. On comparing with the the factored sub-space based i-vector, the proposed decimation approach is noted to have much lower computational complexity and memory requirements but with a small degradation in performance. With this, the proposed decimation approach is very attractive for limited resource platforms due to its low computational burden. Further, in context of distributed speaker verification systems, it can also lead to a very simple synchronization between the client and the server.

5.5 Multi-offset decimation diversity based SV system

Note that, the decimation based projection approach does not take into account the information in all the dimensions of the supervector unlike the other projection methods discussed. In spite of that, it has resulted in a performance quite competitive to that of the simplified i-vector case. This motivated us to build systems using the other dimensions of the supervector by choosing offset-factors other than zero. Table 5.6 shows the performances (averaged over the five test conditions) for all the four offset factors possible for the decimation factor 4 considered. Interestingly, all these systems are found to have very close performances despite each being based on different dimensions of the supervectors. To exploit the complimentary information in these systems, a logistic regression based score level fusion among them is explored as illustrated in Figure 5.5. The resulting system is referred to as *multi-offset decimation diversity based SV* (MODD-SV) system. The scale and shift parameters for the fusion are optimized using the BOSARIS toolkit [61] on the *dev-trials* scores. The performance for the MODD-SV system is also given in Table 5.6. It is to note that by exploiting the diversity among the different decimated representations, the MODD-SV system achieves a relative improvement of 7% over the default i-vector approach in terms of both $C_{\text{DET-12}}$ and EER.

Table 5.6: Performances (averaged over five evaluation conditions) of the MODD-SV system along with that of the component systems using decimation with different offsets. For ease of comparison, the performances for the systems using the default and the simplified i-vector implementations are also given. Note that the systems corresponding all four offsets employed in the decimation of supervectors have resulted in close performances and a significant improvement is achieved with fusion.

SV system		EER (%)	$C_{DET\text{-}12}$
i-vector based	Default	**5.26**	**0.514**
	Simplified	6.20	0.569
Decimated supervector based	Offset-0	6.29	0.587
	Offset-1	6.32	0.589
	Offset-2	6.38	0.594
	Offset-3	6.23	0.572
	MODD (fusion)	**4.93**	**0.478**

In the MODD-SV system, the scores for the component systems can be computed in parallel. Thus, it possess the same run-time as that of the component decimation systems, but the memory usage is increased by a factor equal to the employed decimation factor. In this work, the total memory requirement for the MODD-SV system with four decimation components turns out to be 96 MB which still remains competitive to that of the efficient i-vector approaches reported in the literature. Thus, the proposed MODD-SV system provides an improved performance with a much lower computational complexity and a competitive memory requirement even compared to the efficient implementation of the i-vector based SV system.

5.6 Summary

In this work, the use of a few low-complexity data-independent projections for deriving compact speaker representations from GMM mean supervectors for speaker verification is presented. The low-complexity data-independent projections derived using sparse random matrix and decimation process are explored and contrasted with the existing i-vector based data-dependent ones. The performance for the proposed projections is found to be slightly inferior when compared to the default i-vector implementation and competitive to the fast i-vector implementation, on the NIST 2012 SRE primary task. In terms of computational complexity and memory usage the proposed approaches, especially the decimation based one, are found to have significant advantage even over the fast i-vector implementation. Further, the diversity among the representations obtained by decimation with different offsets is also exploited through score-level fusion. The resulting system

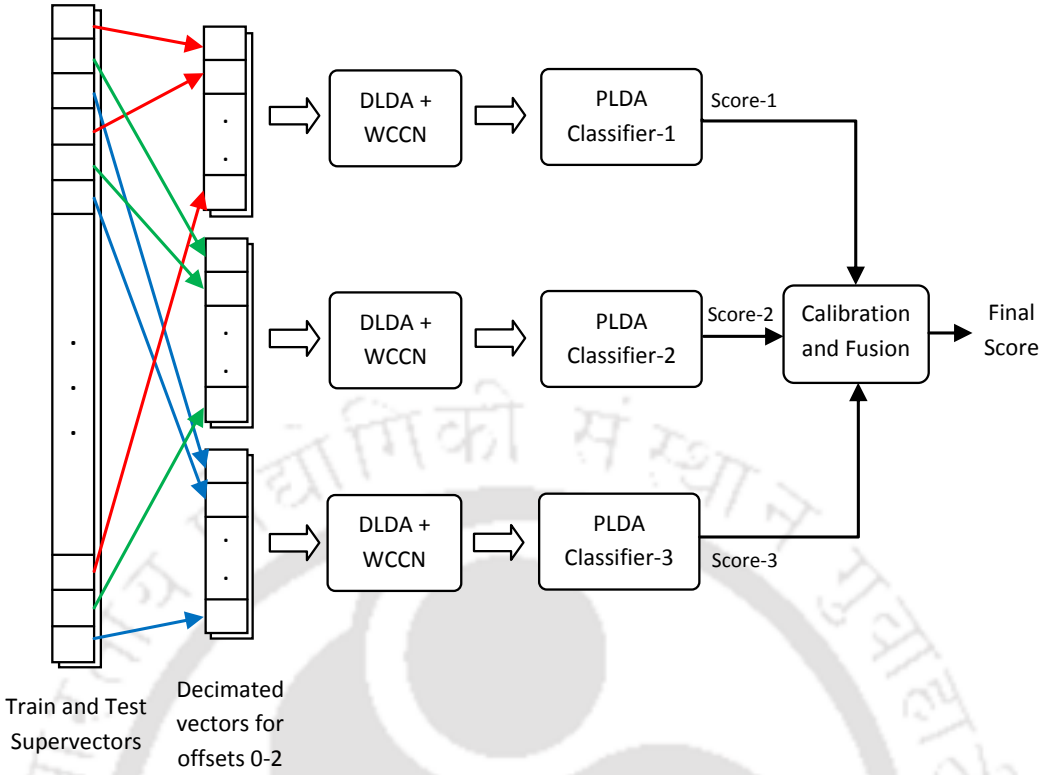


Figure 5.5: Multi-offset decimation diversity based SV system. For ease in depiction *decimation factor 3* case is illustrated.

is noted to provide 7 % relative improvement over the default i-vector based system in terms of both the detection cost and the equal error rate. The computational requirements for the final system remains much lower than that of the default i-vector based and comparable to that of the fast i-vector based systems.

6

Conclusions

Contents

6.1 Summary of the work

There are two major goals that are addressed in this thesis. The first one is to extend the existing sparse representation over exemplar dictionary based speaker verification approach and the other is to explore a few low-complexity data-independent projections in the front-end of the SV system for reducing the overall system complexity.

The thesis begins with the review of the existing speaker verification approaches. It is followed by the description of the recently proposed sparse representation over exemplar dictionary based SV approach. In this approach, a suitable vector representation of a given test utterance is coded as a sparse linear combination of the training examples. For the same, an exemplar dictionary is created by concatenating the vectors representing the training examples of the target (claimed) speaker and those of a set of imposter speakers. It is reasonable to assume that a test vector would lie in the linear span of its own class examples. Thus, in an ideal condition, for a true claim the coefficients of the sparse code corresponding to the target vectors in the dictionary would only be non-zero. On the other hand, for a false claim no such behavior is expected. Exploiting this nature of the sparse representation computed for the true and the false claims, the score for verifying a claim can be obtained with the help of a suitable metric. The salient shortcomings of this approach are discussed which provided the motivation to the thesis work. It is then followed by description of a typical SV system. The salient techniques involved in different modules of an SV system are reviewed, in brief. The details of the concurrent SV approaches involving the GMM mean supervector and the i-vector based speaker modeling are then outlined. These approaches provide the contrast to the sparse representation based approach developed in this work. The various session/channel compensation methods used along with the proposed and the contrast systems are also detailed.

To enhance the generalizability and robustness of the exemplar dictionary for sparse representation, we have proposed the use of a suitably learned dictionary for sparse coding purpose. Unlike the exemplar dictionary, the columns of the learned dictionary are not speaker-labeled and hence the verification can not be performed just with sparse coding the test vector only. To overcome this constraint, an additional sparse coding of the available training examples of the claimed speaker is performed over the same dictionary. The similarity between the sparse codes corresponding to the training and the test vector is measured with the help of the cosine distance to provide the

scores for verification.

The proposed approach of speaker verification using sparse representation has been evaluated using the NIST 2003 SRE dataset. For learning the dictionary, both simple and discriminative approaches are explored. The simple dictionary is derived using the well-known KSVD algorithm whereas for the discriminative one, a supervised variant of KSVD (SKSVD) is used. With the use of a learned dictionary, the sparse representation based SV system is found to outperform the exiting exemplar dictionary based approach significantly. In particular, the discriminatively learned dictionary based system has shown a large margin of improvement and thus resulting in a performance that is even better than that of the i-vector CDS based contrast system. To analyze the system performance with explicit session/channel compensation, the JFA preprocessing is used for the proposed systems and LDA followed by WCCN is used for the i-vector CDS based contrast system. Unlike the trend noted for the uncompensated case, the simple learned dictionary based system is also found to perform significantly better than the i-vector CDS based contrast system. With session/channel compensation, the proposed discriminatively learned dictionary based system happens to be the best performing one with a relative improvement of 31.6 % in terms of EER and 24.3 % in terms of minimum detection cost over the i-vector CDS based system. These performance improvements are attributed to the exploitation of discrimination in speaker space due to the sparseness in the representation and a better generalization due to the dictionary learning in the proposed system. These results and observations lead to the following conclusions: (i) the sparse representation of supervectors over a learned dictionary is *more sensitive* to the session/channel mismatches in comparison to the i-vector approach, and (ii) with proper compensation of these mismatches, the sparse representation over learned dictionary based approach becomes highly effective.

The explicit session/channel compensation using JFA pre-processing is noted to affect the subsequent dictionary learning process and also adds to overall complexity of the system. In addition to these, the JFA approach is also reported to suffer from the loss of some speaker information while discarding the channel factors. Motivated by these facts, a novel approach has been devised that incorporates the session/channel compensation in the sparse coding stage itself. The proposed approach uses two dictionaries, one representing the speaker subspace and the other representing the channel subspace. These dictionaries are learned in supervised manner using labeled development data. The sparse coding of the training and test data supervectors are

computed over the speaker and channel dictionaries jointly and the sparse codes corresponding to the speaker subspace are used for verification using the CDS method. The joint sparse coding based as well as the earlier proposed single dictionary based systems are trained and evaluated using the NIST 2012 SRE dataset. Apart from the sparse representation over the exemplar dictionary based and the i-vector CDS based systems, state-of-the-art i-vector PLDA based system has also been used for contrasting the proposed approaches. The performances of the various systems developed are analyzed primarily in terms of a decision cost function that is used as the primary performance measure in the NIST 2012 SRE. On comparing the performances, it is noted that the sparse representation over simple and discriminative dictionary based systems as well as the i-vector CDS based system follows the same trend as that observed with the earlier experiments done on the NIST 2003 SRE dataset. The joint sparse coding based approach not only avoids the pre-processing stage for session/channel compensation but also achieves an improvement in performance compared to the learned dictionary based system with explicit session/channel compensation using JFA. The possible reasons for the observed improvement are also discussed. On comparing with the i-vector CDS and i-vector PLDA based systems, the joint sparse coding based system with a discriminative dictionary is noted to achieve relative performance improvements of 26.5 % and 14.4 % in terms of the detection cost measure, respectively.

In general, the sparse representation based systems are found to exhibit a very good false alarm performance with most of the false trials producing null scores. Despite the very good performances observed in terms of the detection cost measure, all the sparse representation based systems irrespective of the kind of the dictionary used have resulted in a very poor EERs on the NIST 2012 SRE dataset. This trend is contradictory to the observations made on the NIST 2003 SRE dataset. The reason for the same has been investigated with the help of the DET curve and the histogram of scores. The sparsity constraint for the various systems has been chosen based on the optimization of the NIST SRE 2012 detection cost measure that is computed at a low-false alarm performance of the system. The EER is an application independent performance measure of the system having no bias towards low/high false alarm rates. For the sparse representation based systems using the chosen sparsity, a few true trials are noted to produce null scores like majority of the false trials. On account of this, the proposed system turned out to have a higher false alarm probability for a given low miss probability which resulted in poorer EERs compared to the contrast system. In other words, the sparse representation based systems achieves a very good

low false alarm performance at the cost of a relatively poor non-low false alarm performance. This particular behavior of the proposed sparse representation based systems make it more suitable for practical authentication systems in comparison to the i-vector based approach. On analyzing further, the proposed SR based system is shown to be more robust compared to the i-vector PLDA based system in the short duration test data condition. The reason of this behavior is attributed to low match in case of false trials being unaffected with the reduction in the test data duration due to the sparsity in the representation for the SR based system in contrast to the non-sparse i-vector based one. On assessing the effect of additive noise, both the proposed and the contrast systems are noted to exhibit a similar robustness. The computational complexity of the proposed systems is quite low when compared to the contrast system using the default i-vector extraction and is comparable even to the fastest i-vector extraction methods reported in literature.

The factor-analysis based dimensionality reduction of GMM mean supervectors is the most widely accepted method for deriving the compact representation for speaker verification. The i-vector representations derived using this approach, despite resulting in very good SV performance, suffers from a number of drawbacks. The conventional i-vector extraction algorithm not only has high computational complexity but also requires a large memory space to store the algorithm specific variables as well as the total variability matrix. In addition to that a large volume of data is required for learning the total variability matrix. Data-independent transformation using random matrix is a well-known alternative for the data-dependent methods like PCA and factor analysis in dimensionality reduction. The idea of random data-independent projections are supported by the *Johnson Lindenstrauss (JL)* lemma which states that the projections on a random lower dimensional subspace can the distances between vectors are preserved with a small error. The major advantage of such projections is that it does not require any data-dependent learning. In addition, there exist a few sparse random matrices which lead to faster computation of the projections and require lesser storage space compared to the normal random projection matrix. Motivated by these facts, to reduce the computational complexity in the front-end of a PLDA based SV system, we explored the use of a few data-independent projection approaches instead of factor-analysis for the dimensionality reduction of GMM supervectors. In addition to the sparse random matrix based projection, a simple decimation process that has even lesser complexity, is also explored. Interestingly systems based on both of these projection approaches have resulted in only a small degradation of 0.075 and 0.019 in detection cost compared to the default and a

simplified i-vector based system, respectively. Further considering the computational costs, the decimation based system achieves a 45 fold reduction in run-time and a 13 fold reduction in the memory usage when compared with the simplified i-vector based system. We have also proposed a novel SV system that exploits the diversity among the representations obtained by using different offsets in the decimation of supervector. The resultant system is found to achieve a relative performance improvements of 7 % in detection cost over the default i-vector based system while having a complexity comparable to that of the simplified i-vector based system.

6.2 Summary of contributions

The salient contributions made in the thesis are summarized below.

- A novel SV system based on sparse representation of GMM supervectors over a learned dictionary has been proposed. With the use of proper session/channel compensation, the proposed approach is found to outperform both the existing sparse representation over exemplar dictionary based and the i-vector CDS based SV systems.
- With the motivation to avoid the explicit session/channel compensation, an SV approach using joint sparse coding over learned speaker and channel dictionaries has been developed. With the detailed analysis of results, the proposed sparse representation based system is shown to be a better candidate for high security applications as it provides enhanced low-false alarm performance compared to state-of-the-art i-vector PLDA based system.
- With a motivation to reduce the high computational complexity in SV systems employing compact representations derived from GMM supervectors, the successful use of data-independent front-end projections is demonstrated. The SV systems based on these low-complexity projections have found to perform quite competitive to the i-vector PLDA based one.
- Explored the decimation of GMM supervectors as a non-random data-independent projection approach having very low complexity. A novel SV system that exploits the diversity among the representations obtained by using different offsets in the decimation of GMM supervector, is proposed.
- The various proposed systems are evaluated on a large multi-variability dataset (NIST 2012 SRE) that includes test utterances of variable durations and with ambient noise.

6.3 Conclusions and future directions

This work highlights the pros and cons of the existing i-vector and sparse representation based speaker verification systems and demonstrates the successful exploitation of the merits of both of these approaches. The explicit modeling of the clusters in the speaker space which lead to the achievement of a meaningful sparse representation of speakers is demonstrated and experimentally shown to be a better alternative to the unified modeling followed in the factor analysis based representations. The work also emphasizes that the session/channel compensation through joint sparse coding leads to a lesser loss of speaker information in comparison to the two-stage approach for sparse representation based systems. This enhancement is attributed to the dynamic selection of atoms from the speaker and channel dictionaries as against fixed bases used in the joint factor analysis case. The low-complexity data-independent projections explored in this work are found to be suboptimal in performance compared to i-vector and its simplifications despite offering a significant reduction in complexity. We think the real innovation in that part of the work lies in the development of a multi-offset decimation diversity based SV approach which interestingly provides significant improvement in performance even when compared with the default i-vector based system while still retaining an edge in overall complexity.

Despite the success of the proposed learned dictionary based approach, it is based on a simplifying assumption that all speakers lie in a fixed size subspace and therefore fixed sparsity values are enforced while learning the dictionary and during sparse coding. But such an assumption is definitely far from reality and different speakers are expected to lie in the subspaces having different dimensionality. In literature, we come across some sparse coding algorithms which happen to take into account any possible block structure in the sparse coding [102]. Motivated by that recently block-sparse dictionary learning algorithm is also developed [103]. In this approach, given the data, first the blocks are identified in an unsupervised manner using a sparse agglomerative clustering (SAC) algorithm while the dictionary is learned using block-KSVD (BKSVD) algorithm. So it would be interesting to explore the BKSVD learned dictionary for better generalization of the developed approach, as an immediate future work.

The MODD-SV approach can also be extended to the i-vector domain by the decimation of the statistics collected for computing the i-vector. The different i-vectors so computed from the decimated statistics, can be then used to derive scores which can be fused to generate the final

decision as already discussed. Though such an extension would not be straight forward as the statistics are collected mixture-wise in the default algorithms. The computational complexity of this decimated i-vector SV system would definitely be much lower due to the reduction in size of the T-matrices learned. It is easy to assess that the complexity of such a system can not match that of the MODD-SV approach. But motivated by the large improvements noted with MODD-SV over the single decimation based SV system, we expect a similar trend for the i-vector case too.

The sparse representation over learned dictionary based SV approach developed in this work is generic enough to be applied to other speech based tasks. We have already explored this approach for language identification [104, 105] and emotion recognition [106].

A

Sparse coding algorithms: OMP and CSSOMP

Contents

A.1 Orthogonal matching pursuit

Given a target vector $\boldsymbol{y}$, the dictionary $\boldsymbol{D}$, the orthogonal matching pursuit (OMP) algorithm computes the solution $\boldsymbol{x}$ for the sparse coding problem given as,

$$\min \|\boldsymbol{x}\|_0 \text{ such that } \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} \tag{A.1}$$

OMP is a greedy algorithm and it solves the sparse coding problem given in Eqn. (A.1) iteratively with a constraint either on the sparsity or on the representation error [20, 68]. The steps involved in the OMP algorithm with the use of sparsity constraint are given in the Algorithm 2.

Algorithm 2 Orthogonal matching pursuit

Inputs: Dictionary $\boldsymbol{D} \equiv \{\boldsymbol{d}_p\}_{p=1}^{K}, \boldsymbol{d}_p \in \mathcal{R}^n$, the target vector $\boldsymbol{y}$ and the required number of non-zero coefficients in the sparse solution (sparsity constraint), L.
Initialization: Set the iteration variable $k = 0$, the initial residual $\boldsymbol{r}^0 = \boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}^0 = y$ and the initial solution support $\mathcal{S}_0 = Support\{\boldsymbol{x}^0\} = \emptyset$, the null set.
Main iteration: Increment k by 1 and perform the following steps

(i) **Sweep:** Find the index λ_k that solves

$$\lambda_k = \underset{p=1,\ldots,K}{\arg\max} \left\{ \| < \boldsymbol{r}^{k-1}, \boldsymbol{d}_p > \| \right\} \tag{A.2}$$

(ii) **Update Support:** $\mathcal{S}_k = \mathcal{S}_{k-1} \cup \{\lambda_k\}$, $\boldsymbol{\Phi}_k = [\boldsymbol{\Phi}_{k-1}, \boldsymbol{d}_{\lambda_t}]$.

(iii) **Update Provisional Solution:** Compute the new non-zero coefficients of the solution as,

$$\boldsymbol{x}^k = \underset{\boldsymbol{x}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{\Phi}_k \boldsymbol{x}\|_2 = (\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k)^{-1} \boldsymbol{\Phi}_t^T \boldsymbol{x} \tag{A.3}$$

(iv) **Update Residual:**

$$\boldsymbol{r}^k = \boldsymbol{y} - \boldsymbol{\Phi}_t \boldsymbol{x}^k \tag{A.4}$$

(v) **Stopping Rule:** If $k = L$, stop. Otherwise, set $k = k + 1$ and go back to (i).

Output: Solution $\hat{\boldsymbol{x}}$ is a vector having L non-zero coefficients given by $\boldsymbol{x}^L$ at the indices given by the support $\mathcal{S}^L$

The representation error can also be used as the stopping rule for the OMP algorithm. In such cases, the algorithm is stopped once the representation error r^k reaches the value of the error threshold ϵ_0.

A.2 Class supervised simultaneous OMP

The class supervised simultaneous OMP (CSSOMP) algorithm [79] exploits the internal structure of vectors those belongs to same class and enhances the separability between the sparse codes computed using vectors of different class. In order to achieve this, as shown in Algorithm 3, CSSOMP performs simultaneous sparse coding of the vectors of same class and uses an additional discriminative term along with the reconstruction term in the optimization formulation.

Algorithm 3 Class supervised simultaneous OMP

(i) **Inputs:** Dictionary $\boldsymbol{D} \equiv \{\boldsymbol{d}_p\}_{p=1}^{K}, \boldsymbol{d}_p \in \mathcal{R}^n$, the set of class labeled target vectors $\boldsymbol{Y} \equiv \{\{\boldsymbol{y}_i^j\}_{i=1}^{n_j}\}_{j=1}^{C}$ and the required number of non-zero coefficients in the sparse solution (sparsity constraint), L.

(ii) **Initialization:** Initialize the class counter $q = 1$ and $\Gamma_0 = \emptyset$, the null set.

(iii) **Main iteration:** Selection of L vectors according to the structure of the class q

(a) **Class initialization:** The initial residual $\boldsymbol{r}_i^{j(0)} = y_i^j$, The initial solution support $\mathcal{S}_0^q = \emptyset$, the null set and the iteration counter $k = 1$.

(b) **Sweep:** Find the index λ_k that solves

$$\hat{\boldsymbol{\lambda}}_k^q = \underset{p=1,\ldots,K}{\arg\max} \left\{ \theta.J\left(\left\{ \left\{ < \boldsymbol{r}_i^j(t-1), \boldsymbol{d}_p > \right\}_{i=1}^{n_j} \right\}_{j=1}^{c} \right) - \sum_{i=1}^{n_q} \left\| < \boldsymbol{r}_i^{q,(t-1)}, \boldsymbol{d}_p \right\| \right\}$$

The function $J(.)$ represents the discriminant measure defined as $:= \frac{trace(\boldsymbol{B})}{trace(\boldsymbol{W})}$ where $\boldsymbol{B}$ and $\boldsymbol{W}$ are the *between-class* and the *within-class* covariance matrices of the learning data $\boldsymbol{Y}$, respectively.

(c) **Update Support:** $\mathcal{S}_k^q = \mathcal{S}_{k-1}^q \cup \{\lambda_k^q\}$, $\boldsymbol{\Phi}_k^q = [\boldsymbol{\Phi}_{k-1}^q, \boldsymbol{d}_{\lambda_k^q}]$.

(d) **Update provisional solution and residual:** Compute the new coefficients of the solution as,

$$\boldsymbol{x}_i^{j,(k)} = \left(\boldsymbol{\Phi}_k^{qT} \boldsymbol{\Phi}_k^q\right)^{-1} \boldsymbol{\Phi}_k^{qT} \boldsymbol{x}_i^j$$
$$\boldsymbol{r}_i^{j,(k)} = \boldsymbol{y}_i^j - \boldsymbol{\Phi}_t^q \boldsymbol{x}_i^{j,(k)}$$

(e) **Class stopping rule:** If $k = L$, stop. Otherwise, set $k = k + 1$ and go back to (a).

(f) **Class outputs:** For the vectors in class q the coefficients of the sparse representation are $\boldsymbol{x}_i^{q,(L)}$ at indices given by the support $\mathcal{S}_L^q$

(iv) **Output:** If $q \leq c$ go back to (iii) else **Save** $\boldsymbol{x}_i^{q,(L)} \forall i \in \{1, \ldots, n_q\}$.
$\Gamma_q = \Gamma_{q-1} \cup \mathcal{S}_L^q, q = q + 1$ and **Stop**

B

Dictionary learning algorithms: KSVD, SKSVD and LC-KSVD

Contents

B.1 Dictionary learning with KSVD

Given a set of data examples and a constraint on sparse representations, the KSVD algorithm [62] trains a redundant dictionary by optimizing the following problem:

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}} = \arg\min_{\boldsymbol{D},\boldsymbol{U}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{U}\|_2^2 \quad \text{such that} \quad \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i \tag{B.3}$$

where $\boldsymbol{D}$ is the dictionary of k atoms, $\boldsymbol{U} \equiv \{\boldsymbol{u}_i\}_{i=1}^N$ is the matrix of sparse vectors and l is the chosen sparsity constraint. The KSVD is an iterative algorithm having three stages, viz.the dictionary initialization, the sparse coding, and the dictionary update stages. Dictionary initialization is done either using randomly chosen data vectors from the given set or by generating random numbers. In the sparse coding stage the data vectors are represented as a sparse linear combination of the dictionary. Usually the OMP algorithm is used for performing the sparse coding and the given sparsity constraint of error threshold is used for stopping the OMP algorithm. In the dictionary update stage the atoms of the dictionary are modified such that the overall representation error of

Algorithm 4 Dictionary learning with KSVD

Inputs: Data vectors $\boldsymbol{Y} \equiv \{\boldsymbol{y}_i\}_{i=1}^N$, Number of non-zero coefficients for the sparse solutions (sparsity constraint), L.
Initialization: Initialize the dictionary, $\boldsymbol{D}^0 \in R^{n \times K}$ with l_2 normalized columns. Set the iteration variable $k = 0$.
Main iteration: Increment j by 1 and perform the following steps

(i) **Sparse coding:** Compute sparse codes $\boldsymbol{x}_i$ corresponding to each data vector $\boldsymbol{y}_i$, with a sparsity constraint L, using a pursuit algorithm as,

$$\hat{\boldsymbol{x}}_i = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 \text{ such that } \|\boldsymbol{x}_i\|_0 = L \tag{B.1}$$

(ii) **Dictionary update:** For each column $k = 1, 2 \ldots, K$ in $\boldsymbol{D}^{(j-1)}$, update it by

- Define the group of examples that use this atom, $\omega_k = \{i | 1 \leq i \leq N, \boldsymbol{x}_T^k(i) \neq 0\}$.
- Compute the overall representation error matrix $\boldsymbol{E}_k$ by

$$\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{x}_T^j \tag{B.2}$$

- Restrict $\boldsymbol{E} - k$ by choosing only the columns corresponding to ω_k, and obtain $\boldsymbol{E}_k^R$.
- Apply SVD decomposition $\boldsymbol{E}_k^R = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T$. Choose the updated dictionary column $\boldsymbol{d}_k$ to be the first column of $\boldsymbol{U}$. Update the coefficient vector $\boldsymbol{x}_R^k$ to be the first column of $\boldsymbol{V}$ multiplied by $\Delta(1,1)$

(iii) Set $J = J + 1$

Algorithm 5 Supervised dictionary learning with SKSVD

Inputs: Data vectors $\boldsymbol{Y} \equiv \left\{\{\boldsymbol{y}_i\}_{i=1}^{n_j}\right\}_{j=1}^{c}$, Number of non-zero coefficients for the sparse solutions (sparsity constraint), L.
Initialization: Initialize the dictionary, $\boldsymbol{D}^0 \in R^{n \times K}$ with l_2 normalized columns. Set the iteration variable $k = 0$.
Main iteration: Increment j by 1 and perform the following steps

(i) **Sparse coding:** Compute discriminative sparse codes $\boldsymbol{X} \equiv \left\{\{\boldsymbol{x}_i\}_{i=1}^{n_j}\right\}_{j=1}^{c}$ corresponding to each data vectors in $\boldsymbol{Y}$, with a sparsity constraint L, using CSSOMP,

(ii) **Dictionary update:** For each column $k = 1, 2. \ldots, K$ in $\boldsymbol{D}^{(j-1)}$, update it by

- Define the group of examples that use this atom, $\omega_k = \{i | 1 \leq i \leq N, \boldsymbol{x}_T^k(i) \neq 0\}$.
- Compute the overall representation error matrix $\boldsymbol{E}_k$ by

$$\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{x}_T^j \tag{B.4}$$

- Restrict $\boldsymbol{E} - k$ by choosing only the columns corresponding to ω_k, and obtain $\boldsymbol{E}_k^R$.
- Apply SVD decomposition $\boldsymbol{E}_k^R = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T$. Choose the updated dictionary column $\boldsymbol{d}_k$ to be the first column of $\boldsymbol{U}$. Update the coefficient vector $\boldsymbol{x}_R^k$ to be the first column of $\boldsymbol{V}$ multiplied by $\Delta(1,1)$

(iii) Set $J = J + 1$

the given set of data vectors is minimized. The steps involved in the KSVD algorithm are given in the Algorithm 4

B.2 Supervised dictionary learning with SKSVD

The supervised KSVD (SKSVD) [79] is a modified version of the KSVD for learning discriminative dictionaries using class labeled training data. In SKSVD the sparse coding stage is performed by CSSOMP instead of OMP that is used in KSVD. The steps involved in SKSVD algorithm is described in Algorithm 5

B.3 Supervised dictionary learning with LC-KSVD

The label constrain KSVD (LC-KSVD) algorithm [85, 86] uses a label consistent constraint called discriminative sparse-code error along with the reconstruction error and a constraint on sparsity is used in the optimization for learning the dictionary. Using this combined optimization, the algorithm learns the dictionary and a linear transformation that maps the raw sparse codes to

a more discriminative ones. The objective function used for the learning process is as follows,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{R}} = \underset{\boldsymbol{D},\boldsymbol{U},\boldsymbol{R}}{\arg\min} \left\{ \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{U}\|_2^2 + \alpha \|\boldsymbol{S} - \boldsymbol{R}\boldsymbol{U}\|_2^2 \right\} \quad \text{such that } \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i \tag{B.5}$$

where $\boldsymbol{D}$ is the learned dictionary, $\boldsymbol{X}$ is the set of training data vectors and $\boldsymbol{U}$ is the set of corresponding sparse codes. α is the regularization parameter and the matrix $\boldsymbol{S}$ contains the required discriminative sparse codes created using the class label information of the data vectors. $\boldsymbol{R}$ is a linear transformation matrix that maps the sparse codes of the training targets to a more discriminative ones. For the optimization purpose, the two components of Equation B.5 are rearranged as below,

$$\hat{\boldsymbol{D}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{R}} = \underset{\boldsymbol{D},\boldsymbol{U},\boldsymbol{R}}{\arg\min} \left\| \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\alpha}\boldsymbol{S} \end{pmatrix} - \begin{pmatrix} \boldsymbol{D} \\ \sqrt{\alpha}\boldsymbol{R} \end{pmatrix} \boldsymbol{U} \right\|_2^2 \quad \text{such that } \|\boldsymbol{u}_i\|_0 \leq l \quad \forall i. \tag{B.6}$$

Now, Equation B.6 can be solved using the KSVD algorithm as described in the Section B.1.

Bibliography

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, Jan 2010.

[2] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[3] H. Hermansky, "Perceptual linear predictive (PLP) analysis for speech," *Journal of the Acoustic Society of America*, vol. 87, pp. 1738–1752, 1990.

[4] D. A. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, Jan 1995.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, Jan 2000.

[6] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 203–210, Mar 2005.

[7] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2002, pp. 161–164.

[8] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems*, M. Kearns, S. Solla, and D. Cohn, Eds. MIT Press, Nov 1999, pp. 487–493.

[9] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2006.

[10] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman, "MLLR transforms as features in speaker recognition," in *In Proc. Interspeech*, Sep 2005, pp. 2425–2428.

[11] H. Yang, Y. Dong, X. Zhao, J. Zhao, L. Lu, and H. Wang, "Cluster adaptive training weights as features in SVM-based speaker verification," in *In Proc. Interspeech*, Aug 2007, pp. 2013–2016.

[12] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, May 2006.

[13] C. Longworth and M. J. F. Gales, "Combining derivative and parametric kernels for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 748–757, May 2009.

[14] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Mar 2005, pp. 629–632.

[15] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *In Proc. International Conference on Spoken Language Processing (ICSLP)*, Sep 2006, pp. 1471–1474.

[16] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1448–1460, May 2007.

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

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

[19] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *In Proc. Interspeech*, Aug 2011, pp. 249–252.

[20] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer, 2010.

[21] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Advances in Neural Information Processing Systems*. MIT Press, Dec 2007, pp. 609–616.

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

[23] I. Naseem, R. Togneri, and M. Bennamoun, "Sparse representation for speaker identification," in *In Proc. International Conference on Pattern Recognition (ICPR)*, Aug 2010, pp. 4460–4463.

[24] J. M. K. Kua, E. Ambikairajah, J. Epps, and R. Togneri, "Speaker verification using sparse representation classification," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2011, pp. 4548–4551.

[25] Haris B C and R. Sinha, "Exploring sparse representation classification for speaker verification in realistic environment," in *In Proc. Centenary Conference, Electrical Engineering, Indian Institute of Science*, Dec 2011.

[26] M. Li, X. Zhang, Y. Yan, and S. Narayanan, "Speaker verification using sparse representations on total variability i-vectors," in *In Proc. Interspeech*, Aug 2011, pp. 2729–2732.

[27] J. M. K. Kua, J. Epps, and E. Ambikairajah, "i-vector with sparse representation classification for speaker verification," *Speech Communication*, vol. 55, no. 5, pp. 707–720, 2013.

[28] V. Boominathan and K. S. R. Murty, "Speaker recognition via sparse representations using orthogonal matching pursuit," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar 2012, pp. 4381–4384.

[29] Q. Wu and L. Zhang, "Auditory sparse representation for robust speaker recognition based on tensor structure," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2008:578612, no. 1, Nov 2008.

[30] Y.-H. Long, L.-R. Dai, E.-Y. Wang, B. Ma, and W. Guo, "Non-negative matrix factorization based discriminative features for speaker verification," in *In Proc. International Symposium on Chinese Spoken Language Processing*, Nov 2010, pp. 291–295.

[31] Y. Zu, "Sentences design for speech synthesis and speech recognition database by phonetic rules," in *In Proc. European Conference on Speech Communication and Technology (Eurospeech)*, Nov 1997, pp. 743–746.

[32] R. Saeidi, A. Hurmalainen, T. Virtanen, and D. A. van Leeuwen, "Exemplar-based sparse representation and sparse discrimination for noise robust speaker identification," in *In Proc. Odyssey: The Speaker and Language Recognition Workshop*, Jun 2012.

[33] D. Garcia-Romero and C. Y. Espy-Wilson, "Joint factor analysis for speaker recognition reinterpreted as signal coding using overcomplete dictionaries," in *In Proc. Odyssey: The Speaker and Language Recognition Workshop*, Jun 2010.

[34] O. Glembek, L. Burget, P. Matjka, M. Karafiat, and P. Kenny, "Simplification and optimization of i-vector extraction," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2011, pp. 4516–4519.

[35] M. Li and S. Narayanan, "Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification," *Computer Speech & Language*, vol. 28, no. 4, pp. 940–958, Jul 2014.

[36] S. Cumani and P. Laface, "Fast and memory effective i-vector extraction using a factorized sub-space," in *In Proc. Interspeech*, Aug 2013, pp. 1599–1603.

[37] S. Kaski, "Dimensionality reduction by random mapping: fast similarity computation for clustering," in *In Proc. IEEE International Joint Conference on Neural Networks*, vol. 1, May 1998, pp. 413–418.

[38] F. Bimbot, J. Bonastreand, C. Fredouille, G. Gravier, I. Chagnolleau, S. Meignier, T. Merlin, J. García, D. Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, Apr 2004.

[39] P. Renevey and A. Drygajlo, "Entropy based voice activity detection in very noisy conditions," in *In Proc. European Conference on Speech Communication and Technology (Eurospeech)*, Sep 2001, pp. 1887–1890.

[40] R. Prasad, H. Saruwatari, and K. Shikano, "Noise estimation using negentropy based voice-activity detector," in *In Proc. The Midwest Symposium on Circuits and Systems (MWSCAS)*, vol. 2, Jul 2004, pp. 149–152.

[41] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013, pp. 7229–7233.

[42] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice-Hall, New Jersey, 2001.

[43] T. Kinnunen, R. Saeidi, F. Sedlak, K. A. Lee, J. Sandberg, M. Hansson-Sandsten, and H. Li, "Low-variance multitaper MFCC features: A case study in robust speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 1990–2001, Sep 2012.

[44] M. Sahidullah and G. Saha, "A novel windowing technique for efficient computation of MFCC for speaker recognition," *IEEE Signal Processing Letters*, vol. 20, no. 2, pp. 149–152, Feb 2013.

[45] C. Y. Espy-Wilson, E. Manocha, and S. Vishnubhotla, "A new set of features for textindependent speaker identification," in *In Proc. International Conference on Spoken Language Processing (ICSLP)*, Sep 2006, pp. 1475–1478.

[46] T. Kinnunen and P. Alku, "On separating glottal source and vocal tract information in telephony speaker verification," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr 2009, pp. 4545–4548.

[47] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and MFCC features for speaker recognition," *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 52–55, Jan 2006.

[48] S. R. M. Prasanna, C. S. Gupta, and B. Yegnanarayana, "Extraction of speaker-specific excitation information from linear prediction residual of speech," *Speech Communication*, vol. 48, no. 10, pp. 1243–1261, Oct 2006.

[49] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3–4, pp. 455–472, Jul 2005.

[50] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2095–2103, Sep 2007.

[51] K. Bartkova, D.L.Gac, D. Charlet, and D. Jouvet, "Prosodic parameter for speaker identification," in *In Proc. International Conference on Spoken Language Processing (ICSLP)*, Sep 2002, pp. 1197–1200.

[52] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *In Proc. European Conference on Speech Communication and Technology (Eurospeech)*, Sep 2001.

[53] F. Soong, A. Rosenberg, L. Rabiner, and B.-H. Juang, "A vector quantization approach to speaker recognition," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 10, Apr 1985, pp. 387–390.

[54] K. Farrell, R. Mammone, and K. Assaleh, "Speaker recognition using neural networks and conventional classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 194–205, Jan 1994.

[55] B. Yegnanarayana and S. Kishore, "AANN: an alternative to GMM for pattern recognition," *Neural Networks*, vol. 15, no. 3, pp. 459–469, 2002.

[56] W. M. Campbell, J. Campbell, D. A. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2–3, pp. 210–229, Apr 2006.

[57] S. C. Yin, R. Rose, and P. Kenny, "A joint factor analysis approach to progressive model adaptation in text-independent speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 1999–2010, Sep 2007.

[58] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Montreal, Tech. Rep. CRIM-06/08-13, 2005.

[59] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, , and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *In Proc. Interspeech*, Brigthon, U.K, Sep 2009, pp. 1559–1562.

[60] S. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *In Proc. IEEE International Conference on Computer Vision (ICCV)*, Oct 2007, pp. 1–8.

[61] "The BOSARIS toolkit, (accessed on 10th Dec 2013)," [Online] www.sites.google.com/site/bosaristoolkit/.

[62] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[63] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun 2006, pp. 895–900.

[64] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1061–1073, Sep 2011.

[65] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, Jan 2009.

[66] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio inpainting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 922–932, Mar 2012.

[67] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[68] Y. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *In Proc. Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov 1993, pp. 40–44.

[69] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[70] B. E. Trevor, T. J. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2002.

[71] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun 2010.

[72] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, 2001.

[73] E. J. Candes and D. L. Donoho, "Curvelets- A surprisingly effective nonadaptive representation for objects with edges," in *Curves and Surfaces*, C. Rabut, A. Cohen, and L. L. Schumaker, Eds. Vanderbilt University Press, Nashville TN., 2000, pp. 105–120.

[74] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, Dec 2005.

[75] K. Engan, S. Aase, and H. J. Hakon, "Method of optimal directions for frame design," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Mar 1999, pp. 2443–2446.

[76] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Mar 2005, pp. 293–296.

[77] R. Vidal, Y. Ma, and S. S. Sastry, "Generalized principal component analysis (GPCA)," in *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun 2003, pp. 621–628.

[78] D. I. Moody, S. P. Brumby, J. C. Rowland, and C. Gangodagamage, "Undercomplete learned dictionaries for land cover classification in multispectral imagery of arctic landscapes using CoSA: Clustering of sparse approximations," in *In Proc. Conference Series of Society of Photo-Optical Instrumentation Engineers (SPIE)*, May 2013.

[79] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," IMA Preprint 2213, Univ. of Minnesota, Tech. Rep. 2213, 2008.

[80] M. Li and S. Narayanan, "Robust talking face video verification using joint factor analysis and sparse representation on gmm mean shifted supervectors," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2011, pp. 4835–4838.

[81] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[82] "Nist 2003 speaker recognition evaluation plan," [Online] www.itl.nist.gov/iad/mig/tests/sre/2003/2003-spkrec-evalplan-v2.2.pdf.

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

[84] "Joint factor analysis matlab demo," [Online] www.speech.fit.vutbr.cz/software/joint-factor-analysis-matlab-demo.

[85] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2011, pp. 1697–1704.

[86] Z. Jiang, Z. Lin, and L. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, Nov 2013.

[87] "The NIST year 2012 speaker recognition evaluation plan," [Online] www.nist.gov/itl/iad/mig/upload/ NIST_SRE12_evalplan-v17-r1.pdf.

[88] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech," in *In Proc. Odyssey: The Speaker and Language Recognition Workshop*, Jun 2010.

[89] Haris B C, G. Pradhan, R. Sinha, and S. R. M. Prasanna, "The IITG speaker verification systems for NIST SRE 2012," in *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013, pp. 7668–7672.

[90] O. Glembek, L. Burget, N. Brummer, O. Plchot, and P. Matjka, "Discriminatively trained i-vector extractor for speaker verification," in *In Proc. Interspeech*, Aug 2011, pp. 137–140.

[91] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "i-vector based speaker recognition on short utterances," in *In Proc. Interspeech*, Aug 2011, pp. 2341–2344.

[92] V. Hautamaki, Y.-C. Cheng, P. Rajan, and C.-H. Lee, "Minimax i-vector extractor for short duration speaker verification," in *In Proc. Interspeech*, Aug 2013, pp. 3708–3712.

[93] S. Cumani and P. Laface, "Memory and computation trade-offs for efficient i-vector extraction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 934–944, May 2013.

[94] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," *Random Structures and Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[95] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *In Proc. ACM International Conference on Knowledge Discovery and Data Mining*, Aug 2006, pp. 287–296.

[96] D. Achlioptas, "Database-friendly random projections," in *In Proc. ACM Symposium on Principles of Database Systems*, Aug 2001, pp. 274–281.

[97] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, Oct 2001.

[98] "NIST 2005 speaker recognition evaluation plan," [Online] www.itl.nist.gov/iad/mig/tests/spk/2005/sre-05_evalplan-v6.pdf.

[99] "NIST speaker recognition evaluations website," [Online] www.itl.nist.gov/iad/mig/tests/spk/.

[100] P. Kenny, "A small footprint i-vector extractor," in *In Proc. Odyssey: The Speaker and Language Recognition Workshop*, Jun 2012.

[101] M. Li, "Simplified i-vector MATLAB code (accessed on 14th Apr 2014)," [Online] www.jie.sysu.edu.cn/∼mli/SimplifiedSupervisedIvectorMatlab.zip.

[102] Y. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, Jun 2010.

[103] L. Zelnik-Manor, K. Rosenblum, and Y. Eldar, "Dictionary optimization for block-sparse representations," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2386–2395, May 2012.

[104] O. Singh, Haris B C, R. Sinha, B. Chettri, and A. Pradhan, "Sparse representation based language identification using prosodic features for indian languages," in *In Proc. Annual IEEE India Conference (INDICON)*, Dec 2013, pp. 1–5.

[105] O. Singh, Haris B C, and R. Sinha, "Language identification using sparse representation: A comparison between gmm supervector and i-vector based approaches," in *In Proc. Annual IEEE India Conference (INDICON)*, Dec 2013, pp. 1–4.

[106] A. Chauhan and A. Prasad, "Speech based emotion recognition using sparse representation classification," Apr 2012, Bachelor of Technology Project Report, Dept. of EEE, Indian Institute of Technology Guwahati.

List of Publications

Journal Publications

(i) **Haris B C** and R. Sinha, "Low-complexity speaker verification with decimated supervector representations," *Speech Communication*, Elsevier, Vol. 68, pp. 11-22, April 2015.

(ii) **Haris B C** and R. Sinha, "Exploring data-independent dimensionality reduction in sparse representation based speaker identification," *Circuits, Systems & Signal Processing*, Springer, pp. 1-18, April 2014.

Manuscripts Under Revision

(i) **Haris B C** and R. Sinha "Robust speaker verification with joint sparse coding over learned dictionaries," (Under 3^{rd} review in, *IEEE Transactions on Information Forensics and Security*)

Conference and Workshop Publications

(i) **Haris B C**, G. Pradhan, R. Sinha, and S. R. M. Prasanna, "The IITG speaker verification systems for NIST SRE 2012," in *ICASSP-2013, May 2013*

(ii) **Haris B C** and R. Sinha, "On exploring the similarity and fusion of i-vector and sparse representation based speaker verification systems," in *Proc. Odyssey 2012: The Speaker and Language Recognition Workshop*, June 2012.

(iii) **Haris B C** and R. Sinha, "Sparse representation over learned and discriminatively learned dictionaries for speaker verification," in *Proc. ICASSP-2012*, March 2012.

(iv) **Haris B C** and R. Sinha, "Sparse representation of total variability smoothed GMM mean supervectors for speaker verification," in *Proc. International Conference on Signal Processing and Communications, 2012*, July 2012.

(v) **Haris B C** and R. Sinha, "Speaker verification using sparse representation over KSVD learned dictionary," in *Proc. 18th National Conference on Communications 2012*, Feb. 2012.

(vi) **Haris B C** and R. Sinha, "Exploring sparse representation classification for speaker verification in realistic environment," in *Proc. Centenary Conference, Electrical Engineering, Indian Institute of Science, Bangalore*, Dec. 2011.

(vii) O. P. Singh, **Haris B C** and R. Sinha, "Language identification using sparse representation: A comparison between GMM supervector and i-vector based approaches," in *Proc. IEEE Indicon-2013*, Dec. 2013.

(viii) O. P. Singh, **Haris B C**, R. Sinha, B. Chettri and A. Pradhan "Sparse representation based language identification using prosodic features for Indian languages," in *Proc. IEEE Indicon-2013*, Dec. 2013.