PNCC-*i*vector-SRC based Speaker Verification

Eliathamby Ambikairajah[1,2], Jia Min Karen Kua[1], Vidhyasaharan Sethu[1] and Haizhou Li [3,1]
[*] School of Electrical Engineering and Telecommunications,
The University of New South Wales, Sydney, NSW 2052, Australia
[†]ATP Research Laboratory, National ICT Australia (NICTA), Eveleigh 2015, Australia
[^] Human Language Technology, Institute for Infocomm Research (I^2R), Singapore 138632
E-mail: ambi@ee.unsw.edu.au, j.kua@unswalumni.com, v.sethu@unsw.edu.au, hli@i2r.a-star.edu.sg

***Abstract* - Most conventional features used in speaker recognition are based on Mel Frequency Cepstral Coefficients (MFCC) or Perceptual Linear Prediction (PLP) coefficients. Recently, the Power Normalised Cepstral Coefficients (PNCC) which are computed based on auditory processing, have been proposed as an alternative feature to MFCC for robust speech recognition. The objective of this paper is to investigate the speaker verification performance of PNCC features with a Sparse Representation Classifier (SRC), using a mixture of ℓ_1 and ℓ_2 norms. The paper also explores the score level fusion of both MFCC and PNCC *i*-vector based speaker verification systems. Evaluations on the NIST 2010 SRE extended database show that the fusion of MFCC-SRC and PNCC-SRC gave the best performance with a DCF of 0.4977. Further, cosine distance scoring (CDS) based systems were also investigated and the fusion of MFCC-CDS and PNCC-CDS presented an improvement in terms of EER, from a 3.99% EER baseline to 3.55%.**

I. INTRODUCTION

Cepstral features such as Mel-Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictive (PLP) coefficients are commonly used features in most conventional speaker verification systems. These features have shown remarkable performance for speaker verification in matched channel conditions [1]. However, when the test environment is different from the training environment, the observed features mismatch and yield poor recognition performance. The need to find a set of features that are robust with respect to channel variability and has limited degradation in performance led to the recent development of the Power Normalised Cepstral Coefficients (PNCC) feature by Kim et al for speech recognition. The use of PNCC features provided a significant improvement in speech recognition accuracy compared to MFCC in the presence of additive noises and in reverberant environments [2]. Inspired by the findings that PNCC features address the robustness issue well with respect to acoustic variability, we investigate the use of PNCC features in speaker verification where acoustic variability remains one of the key challenges.

Recently, the discriminative abilities of sparse representation classification have been exploited for speaker verification and recognition tasks [3, 4] using Gaussian Mixture Model (GMM) supervectors, whereas Li et al [5] used total variability *i*-vectors and demonstrated that there was an overall improvement in performance when fusing the *i*-vector based Support Vector Machine (SVM) with the Cosine Distance Scoring (CDS) system. The total variability *i*-vector modeling has received significant attention due to its remarkable performance [6, 7].

Haris et al [8] used learned dictionaries with the K-SVD algorithm, instead of the more conventional exemplar dictionary based SRC, for speaker verification tasks. The learned dictionary based SRC was more data independent and provided an improved performance over exemplar dictionaries when evaluated on a smaller database (NIST2003).

In this paper, the authors will look at developing a speaker verification system implemented with the following 3 key stages: (i) PNCC features, (ii) GMM supervectors converted to *i*-vectors, (iii) using an exemplar dictionary based SRC system. The paper will be divided into the following sections: Section II will look at a speaker verification framework. Section III will cover the extraction of PNCC features and then Section IV will explain how these features are used with a SRC. The final section of the paper will look at and evaluate the results of fused and individual speaker verification systems.

II. SPEAKER VERIFICATION FRAMEWORK

A speaker verification system comprises many stages, as shown in Fig 1. These stages include the feature extraction, feature normalization, speaker modeling, speaker model compensation, classification (scoring), score normalization and finally, a decision process, and the overall proposed system will be referred to in this paper as the PNCC-*i*vector-SRC speaker verification system.

A. Feature Extraction Stage

Most verification systems use MFCC/PLP, Delta (Δ) MFCC/PLP and Delta-Delta($\Delta - \Delta$) MFCC/PLP as features (e.g. 60 dimensions) as shown in Fig 1. These are followed by a feature normalization/warping stage to improve the robustness of the system to channel effects. The feature normalization stage of the systems reported in this paper consists of feature warping which "warps" the cumulative distribution function of the features to a reference distribution (typically Gaussian distribution) [1]. Commonly used features in speaker recognition/verification systems are outlined in [9]. The Power Normalised Cepstral Coefficients

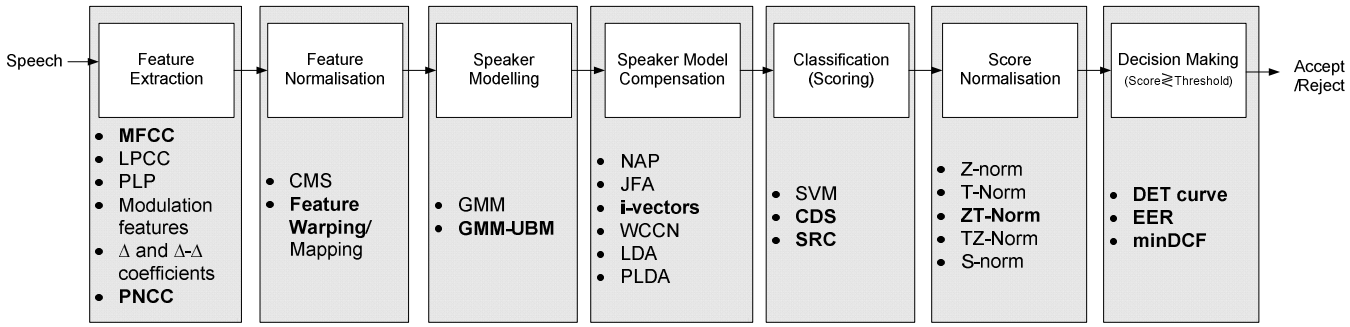

Fig. 1 The stages of a speaker verification system. Aspects of each stage that are investigated in this paper are shown in bold.

(PNCC) feature extraction stage is explained in more detail in section III.

B. Speaker Modeling Stage

Speaker modeling is based on Gaussian mixture models (GMM) which model the probability distributions of the features and are completely parameterised by their mean vectors, covariance matrices and mixture weights. Over the years this speaker modeling approach has evolved into the successful GMM-UBM paradigm where individual speaker models are obtained by adapting the means (typically) of a universal background model (UBM). This then leads to a convenient vectorial representation of an utterance obtained by concatenating the mean vectors obtained by adapting the UBM with the utterance to form a larger vector referred to as a supervector, $\boldsymbol{s}$, which has a dimension of $N \cdot D$ where N is the number of Gaussian mixtures and D is the dimensionality of the feature vector.

For a given speaker, mismatched conditions can occur due to the use of different telephone handsets or different acoustic environments (channels) between the acquired training and test speech utterances. As such, channel (speaker model) compensation at the modeling stage is necessary to make sure that the estimated supervectors do not vary a lot for that given speaker. The difficulties associated with compensating for these differences have presented an active research topic for the speaker verification field in recent years and some of the state-of-the-art channel compensation schemes include the Joint Factor Analysis (JFA) [10] and the *i*-vectors [7].

C. Speaker Model Compensation Stage - JFA

In the speaker model compensation stage of Fig. 1, the *i*-vectors are extracted for use in this paper, but it is valuable to review the JFA method, to gain a better understanding of the context.

The JFA is a powerful method used for compensating the mismatched conditions caused by different channel and inter-session variability. In JFA, the GMM mean supervector $\boldsymbol{s}$, for a given speaker can be represented as the sum of 4 factors as in (1):

$$\mathbf{s} = \boldsymbol{\mu} + \mathbf{U}\boldsymbol{x} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z} \quad (1)$$

where $\boldsymbol{\mu}$ is the speaker-independent and channel-independent supervector, $\boldsymbol{U}$ and $\boldsymbol{V}$ are rectangular matrices representing the principal direction of the speaker and channel variability and $\boldsymbol{D}$ is the diagonal residual matrix. $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ are the speaker, common and channel factors in the JFA model respectively.

Collectively, the matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{D}$ are called the hyper-parameters of the JFA model and are usually estimated beforehand on large labeled development datasets. For a given training sample, the latent factors $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly estimated, followed by the estimation of $\boldsymbol{z}$.

The above classical joint factor analysis modeling based on speaker and channel factors has two distinct spaces: the speaker space defined by the eigen-voice matrix **U** and the channel space defined by the eigen-channel matrix **V**.

D. Speaker Model Compensation Stage – i-vector

Recently, Dehak et al defined a new space, termed the "total variability space", which contains the speaker and channel variabilities simultaneously [11]. In the new model, no distinction between the speaker effects and the channel effects in GMM supervector space is made because experimental work carried out in [11] shows that channel factors estimated using JFA, also contained information about speakers.

For the new model, the new speaker-dependent and channel-dependent GMM supervector for a given utterance, defined in (1) is re-written as follows:

$$\mathbf{s} = \boldsymbol{\mu} + \mathbf{P}\mathbf{q} \quad (2)$$

where $\boldsymbol{\mu}$ is the UBM mean supervector of dimension ND (where N is typically 1024 or 2048 and D maybe around 60), **P** is the total variability matrix and **q** is the identity vector (*i*-vector) of dimension typically around 400. It can be seen that the dimensionality of the supervector has been transformed from a high dimension vector to a low dimensional total variability space.

E. Classification (Scoring) Stage - CDS

The total variability space has 2 channel compensation techniques which are used to reduce the channel variabilities. These techniques are the Linear Discriminant Analysis (LDA) [7] and the Within Class Covariance Normalization (WCCN) [12]. In the LDA, feature vectors are projected to a set of new orthogonal axes, where the discrimination between the different speakers is maximized. This is done via eigen-decomposition, obtaining the projection matrix as the eigenvectors corresponding to the largest eigenvalues obtained from the eigenvalue equations, $(\mathbf{W^{-1}B})\mathbf{v} = \lambda\mathbf{v}$, where $\mathbf{W}$ is the within-class covariance matrix, $\mathbf{B}$ is the between-class covariance matrix. Therefore, LDA can be viewed as, in some sense, maximising the ratio of between-class variance to within-class variance. The WCCN scales the subspace and by doing so, reduces the dimensions with high within-class variance.

In the total variability space, a new classification method based on cosine distance, termed the Cosine Distance Scoring (CDS) classifier, as shown in (3), can then be used for classification, where $\mathbf{q}_{tst}$ and $\mathbf{q}_{trt}$ are the test and target speakers' *i*-vectors respectively and $\langle .,. \rangle$ denotes the inner product.

$$\text{score}(\mathbf{q}_{tst}, \mathbf{q}_{trt}) = \frac{\langle \mathbf{q}_{tst}, \mathbf{q}_{trt} \rangle}{\|\mathbf{q}_{tst}\| \|\mathbf{q}_{trt}\|} \qquad (3)$$

The CDS compares the angles between a channel compensated test *i*-vector and a channel compensated target *i*-vector.

The final two stages in Fig. 1 are score normalization and decision making. Score normalization attempts to remove the effect of noise and channel variability by modifying the score distribution [13-15].

III. PNCC FEATURE EXTRACTION

The PNCC feature extraction algorithm [2] has the same degree of computational complexity as that of the MFCC or PLP coefficient extraction algorithms, but it has many of the attributes of human auditory processing. The PNCC feature extraction algorithm consists of three major stages: (a) Initial processing (b) Environmental processing (c) Final processing.

A. Initial Processing Stage

The initial processing comprises pre-emphasis of the speech signal, followed by Short-Time Fourier Transform (STFT) using a Hamming window of 20ms duration with 10ms of frame overlap. The magnitude of STFT outputs are squared and weighted by a 31-channel gammatone filterbank that covers the bandwidth of telephone speech (300-3400Hz). The center frequencies of the gammatone filters are linearly spaced in the Equivalent Rectangular Bandwidth (ERB) auditory frequency scale. The output of initial processing is a vector $\left[P_1^{(m)}, P_2^{(m)}, \dots, P_N^{(m)}\right]$ of *N*-dimensions (*N*=31) of short-time spectral estimates, corresponding to each gammatone filter, where N is the number of filters and $m = 1, 2, \dots, M$ is the frame number and M is the total number of frames.

B. Environmental Processing Stage

The environmental processing stage has 2 sub-stages as illustrated in Fig. 2: temporal processing and spectral smoothing. The temporal processing is carried out by first estimating a temporal smoothing 'transfer function', $\boldsymbol{S^{(m)}} = \left[S_1^{(m)}, S_2^{(m)}, \dots, S_N^{(m)}\right]$ and then applying it to the short-time spectral estimates $\left[P_1^{(m)}, P_2^{(m)}, \dots, P_N^{(m)}\right]$ by multiplying them. In order to estimate this 'transfer function', first, the short-time spectral power estimate vector $\left[P_1^{(m)}, P_2^{(m)}, \dots, P_N^{(m)}\right]$ is modified by computing the running average (low-pass filtering) over 5 frames, as shown by the vertical rectangular boxes in the spectrogram (Fig. 2) and the new smoothed short-time spectral power estimate vector is termed $\boldsymbol{Q^{(m)}} = \left[Q_1^{(m)}, Q_2^{(m)}, \dots, Q_N^{(m)}\right]$.

Temporal asymmetric low pass filtering is then carried out on the new vectors to estimate the lower envelope which is then removed from $\boldsymbol{Q}$ prior to halfwave rectification and estimation of temporal masking thresholds which is combined with a noise floor estimate to obtain $\boldsymbol{R^{(m)}} = \left[R_1^{(m)}, R_2^{(m)}, \dots, R_N^{(m)}\right]$. The temporal smoothing transfer function, $\boldsymbol{S^{(m)}}$, is then obtained by smoothing the ratio of $\boldsymbol{R^{(m)}}$ to $\boldsymbol{Q^{(m)}}$ with a running temporal average. This process is illustrated in Fig 3.

The short-time spectral estimate $\left[P_1^{(m)}, P_2^{(m)}, \dots, P_N^{(m)}\right]$ corresponding to each gammatone filter is now multiplied by $\left[S_1^{(m)}, S_2^{(m)}, \dots, S_N^{(m)}\right]$ to obtain a new vector $\left[T_1^{(m)}, T_2^{(m)}, \dots, T_N^{(m)}\right]$. This is called the time-frequency normalization vector, as in Fig. 2.

The spectral smoothing sub-stage takes the time-frequency normalization vector $\left[T_1^{(m)}, T_2^{(m)}, \dots, T_N^{(m)}\right]$ and processes it along the frequency axis to obtain a mean power estimate for each frame. This is shown by the horizontal rectangular box in the spectrogram in Fig. 2. Once the mean power estimate for each frame is calculated, the values obtained are then smoothed.

From the time-frequency normalization vector, a running average, $\mu(m)$, along the frequency axis is calculated for that frame and the new vector $\left[U_1^{(m)}, U_2^{(m)}, \dots, U_N^{(m)}\right]$ is obtained by dividing the time-frequency normalization vector by $\mu(m)$. It should be noted that while the MFCC extraction algorithm has stages that are analogous to the initial and final processing stages of the PNCC algorithm it doesn't have one analogous to the environmental processing stage of the PNCC algorithm.

C. Final Processing Stage

In the final processing stage, the vector $\left[U_1^{(m)}, U_2^{(m)}, \dots, U_N^{(m)}\right]$ is weighted by power-law non-linearity,

$\left[\left(U_1^{(m)}\right)^{\frac{1}{p}}, \left(U_2^{(m)}\right)^{\frac{1}{p}}, \ldots, \left(U_N^{(m)}\right)^{\frac{1}{p}}\right]$, where $p = 15$. The resulting vector $\left[V_1^{(m)}, V_2^{(m)}, \ldots, V_N^{(m)}\right]$ is obtained and a Discrete Cosine Transform (DCT) is applied to this vector to obtain the PNCC coefficients $\left[PC_1^{(m)}, PC_2^{(m)}, \ldots, PC_N^{(m)}\right]$, as shown in Fig. 2.

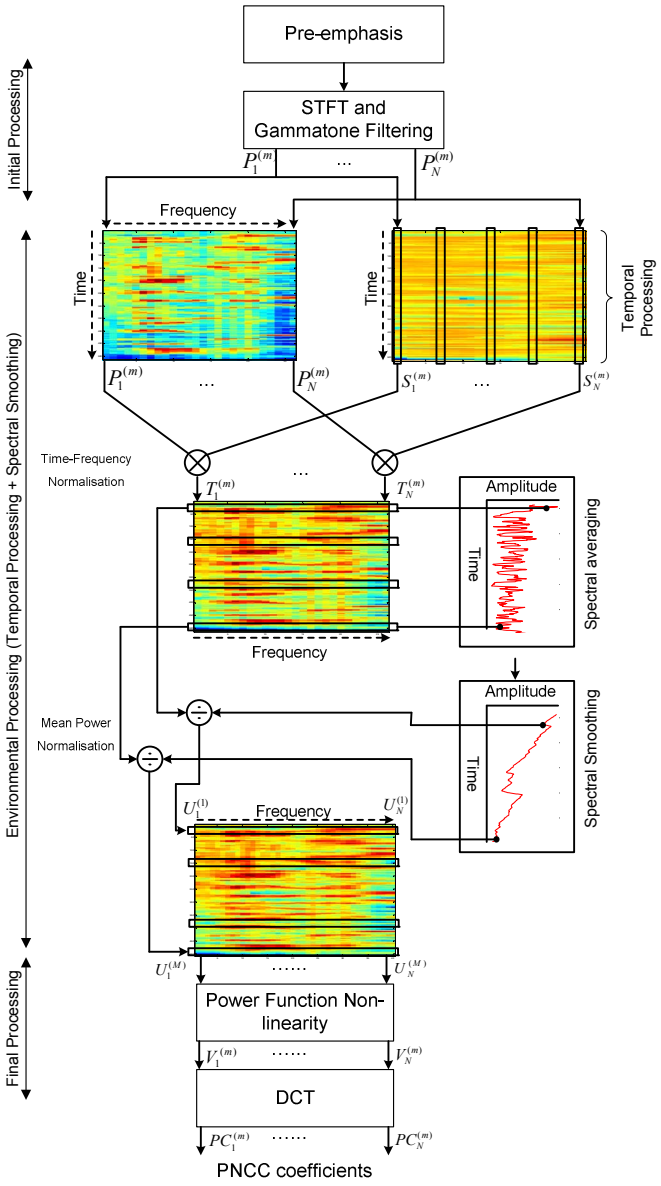

Fig. 2 Structure of the PNCC feature extraction algorithm

IV. SPARSE REPRESENTATION CLASSIFIER (SRC)

In a classification problem, labeled training data feature vectors from different classes are used and then the test data feature vector $\boldsymbol{\nu}$ is assigned to a particular class, using an algorithm. For example, in a speaker verification task, feature vectors are obtained by mapping each variable length training utterance to an i-vector of fixed-dimension $(M \times 1)$ obtained after speaker model compensation of GMM mean supervectors as outlined in section II.

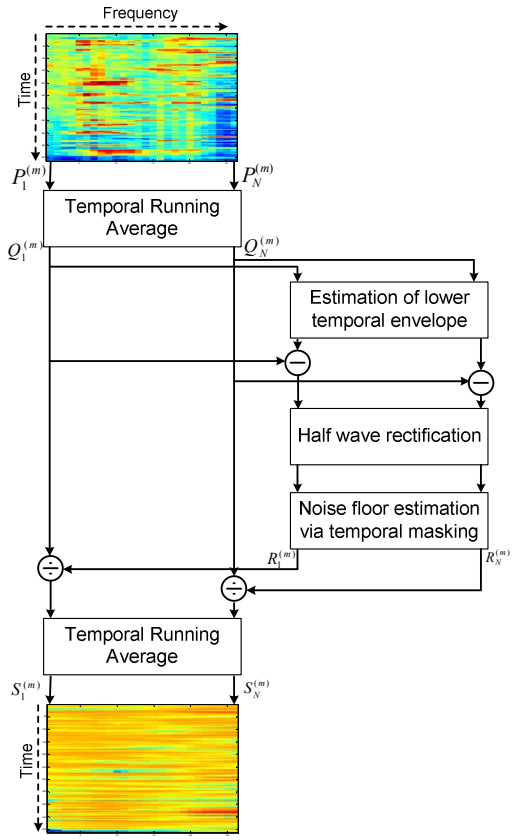

Fig. 3 Temporal processing stage of environmental processing

Assume that all training feature vectors, from the k^{th} speaker are placed in a matrix $\mathbf{A_k}$ as column vectors to serve as exemplars; $\mathbf{A}_k = [\boldsymbol{\nu}_{k1}, \boldsymbol{\nu}_{k2}, \boldsymbol{\nu}_{k3} \ldots \boldsymbol{\nu}_{kN}]$ where $\boldsymbol{\nu}_{kN}$ represents the N^{th} training feature vectors of the k^{th} speaker. If $\mathbf{y}$ is a test feature vector $(M \times 1)$ from the k^{th} speaker, then $\mathbf{y}$ can be represented as a weighted linear combination of all entries in $\mathbf{A}_k$.

$$\mathbf{y} = \alpha_{k1}.\boldsymbol{\nu}_{k1} + \alpha_{k2}.\boldsymbol{\nu}_{k2} + \cdots \alpha_{kN}.\boldsymbol{\nu}_{kN} \tag{4}$$

where α_k are scalar quantities (weights) to be determined. In order to determine the class of $\mathbf{y}$ (i.e. finding the weights), a global exemplar dictionary matrix $\mathbf{A}$ $(M \times L)$ needs to be developed to include training feature vectors from all k classes (speakers) by concatenating $\mathbf{A}_k$ $(k = 1,2, \ldots, K)$.

$$\begin{aligned}\mathbf{A} &= [\ \mathbf{A}_1\ ,\ \mathbf{A}_2\ ,\cdots,\ \mathbf{A}_K\ ] \\ &= [\boldsymbol{\nu}_{11}, \boldsymbol{\nu}_{12}, \cdots, \boldsymbol{\nu}_{1N}, \vdots\ \boldsymbol{\nu}_{21}, \boldsymbol{\nu}_{22}, \cdots, \boldsymbol{\nu}_{2N}, \vdots \cdots \vdots, \boldsymbol{\nu}_{K1}, \boldsymbol{\nu}_{K2}, \cdots, \boldsymbol{\nu}_{KN}]_{MxL} \\ &\quad\ \textit{speaker 1} \qquad \textit{speaker 2} \qquad \textit{speaker k}\end{aligned} \tag{5}$$

where $L = KN$. The number of column vectors in $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \ldots$ depends on the number of training utterances available for each speaker. For simplicity, it is assumed that

all classes have the same number of training utterances (N). Now the test vector $\mathbf{y}$ can be represented as a linear combination of all k classes of training feature vectors ($\mathbf{x}$) using the matrix $\mathbf{A}$ above. i.e

$$[\mathbf{y}]_{Mx1} = [\mathbf{A}]_{MxL}[\mathbf{x}]_{Lx1} \tag{6}$$

The linear system of equations (6) can be solved and the class of $\mathbf{y}$ can be found by using the information in $\mathbf{x}$. For example, if $\mathbf{y}$ belongs to speaker 1, then the weights of $\mathbf{x}$ (i.e α's) that are not associated with the speaker 1 should ideally be zero.

$$\mathbf{x} = [\underbrace{\alpha_{11}, \alpha_{12}, \alpha_{13}, \cdots \alpha_{1N}}_{speaker\ 1}, \vdots\ 0,0,0,0, \cdots, \vdots\ \underbrace{0,0,0,0, \cdots}_{speaker\ k}]^T \tag{7}$$

Ideally the vector $\mathbf{x}$ will exhibits a high level of sparsity, and the non-zero weights will correspond to exemplars from speaker 1. In practice this sparsity condition is enforced by choosing an appropriate solution to the system of linear equations given by (6). The feature vector dimension M is much smaller than L; ($M \ll L$). Therefore the system, $\mathbf{y} = \mathbf{Ax}$, has more unknown ($\alpha's$) than equations (i.e. an underdetermined system) and has infinitely many solutions. In order to obtain a single well-defined solution, and preferably in this case the sparsest solution, additional criteria are needed and this is achieved using the following optimization techniques:

Find the weight vector, $\mathbf{x}$, such that $\mathbf{y} = \mathbf{Ax}$ and $\|\mathbf{x}\|_1$ is minimized. That is

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad subject\ to\ \mathbf{y} = \mathbf{Ax} \tag{8}$$

where $\|\mathbf{x}\|_1$ is the ℓ_1-norm. This will lead to a sub-optimal solution where iterative methods like the matching pursuit and orthogonal matching pursuit are used. A generalized version of (8) is given below:

$$\min_{x} \|\mathbf{y} - \mathbf{Ax}\|_2 + \beta\|\mathbf{x}\|_1 \tag{9}$$

where the vector $\mathbf{y}$ is noisy and assumed to be generated by $\mathbf{y} = \mathbf{Ax} + \boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}$ is a white Gaussian noise vector. The regularization parameter β (where $0 \le \beta \le 1$) controls the weight of the ℓ_1-norm. Equation (9) is known as the LASSO problem [16].

Equation (9) can be modified to impose a mixture of an ℓ_1-norm and ℓ_2-norms constraints on x and is given below:

$$\min_{x} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \beta\|\mathbf{x}\|_1 + (1-\beta)\|\mathbf{x}\|_2^2 \tag{10}$$

Equation (10) is known as the Elastic net problem [16]. The ℓ_1 term enforces the sparsity of the solution and the ℓ_2 penalty has a smoothing effect that stabilizes the obtained solution.

Once the underdetermined linear system $\mathbf{y} = \mathbf{Ax}$ is solved using equation (8) or (9) or (10), the weight vector $\mathbf{x}$ can be used as a new feature extracted from the test vector $\mathbf{y}$ for classification purposes. The new feature vector $\mathbf{x}$, should be as discriminative as possible between many classes (or speakers). Ideally, the new feature vector $\mathbf{x}$ should have non-zero entries associated with the class (speaker) of test vector $\mathbf{y}$ as in (7). The modeling error can cause non-zero values at the entries of $\mathbf{x}$ other those of the class of $\mathbf{y}$. Therefore the contribution of individual class (or speaker) in the dictionary $\mathbf{A}$ is represent the test vector $\mathbf{y}$, should be calculated in terms of residual error for classification purposes.

The residual error (R_k) of the k^{th} class is calculated by retaining the weights associated with that class in the vector $\mathbf{x}$ and setting all the other entries in $\mathbf{x}$ that are not associated with the k^{th} class to zero. The residual error for the k^{th} class is given in (11):

$$\begin{aligned} R_k &= \|\mathbf{y} - \mathbf{Ax}_k\|_2 \\ \mathbf{x_k} &= [\underbrace{0000}_{spk\ 1} \vdots\ 0000 \vdots \cdots \vdots \underbrace{\alpha_{k1}, \alpha_{k2}, \cdots \alpha_{kN}}_{spk\ k}]^T \end{aligned} \tag{11}$$

The residual error can be normalized as follows:

$$R_{kn} = \frac{\|\mathbf{y} - \mathbf{Ax}_k\|_2}{\|\mathbf{y}\|_2} \tag{12}$$

The test vector $\mathbf{y}$ is then assigned to the class that minimizes the normalized residual error R_{kn}.

In order to demonstrate equations (8) – (10), a dictionary $\mathbf{A}$ was created using a number of two-dimensional data, where the columns of matrix $\mathbf{A}$ represented 4 classes, each with 5 training samples. These were labeled as *Class 1* to *Class 4*. The three test vectors namely *Test 1*, *Test 2* and *Test 3* were chosen such that they belonged to Class 1. Fig. 4 shows the clusters formed by each class and the position of the test vectors in the scatter diagram.

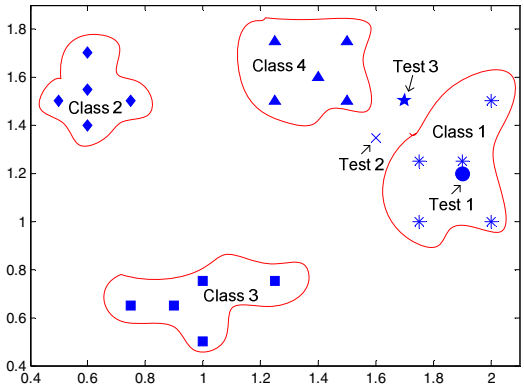

Fig. 4 Plot of entries in matrix $\mathbf{A}$ and three different test vectors $\mathbf{y}$, where class cluster boundaries are just a rough approximation.

In solving (8) – (10) with the residual error computed using (12), a residual error for each class was computed for each sparseness algorithm. Experimentally, ℓ_1-minimization with equality constraints (8), was only able to classify *Test 1*

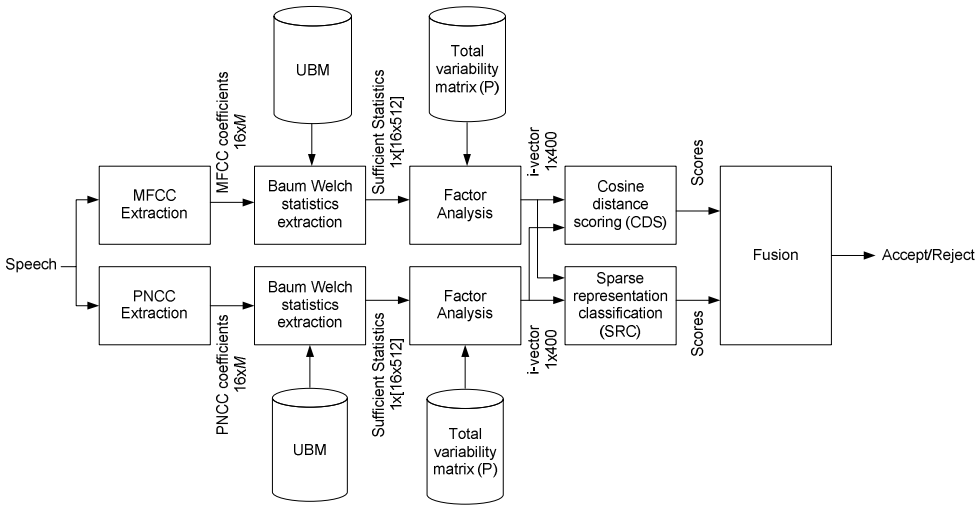

Fig. 5 Fusion of both MFCC and PNCC i-vector based speaker verification systems

correctly, whereas Lasso (9) was able to classify *Tests 1* and *2* correctly. On the other hand, Elastic Net was the only sparseness algorithm (10) that was able to classify all three test samples correctly.

The, ℓ_1 -minimization with equality constraint (8) and Lasso (9) classified *Test 3* as *Class 4*, whereas Elastic Net classified *Test 3* correctly as *Class 1*, as shown in Table 1. This supported the claim in [16] that sparseness techniques that employed a combination of ℓ_1 and ℓ_2 norm (i.e. Elastic Net) offered the best performance in classification tasks.

TABLE 1.
RESIDUAL ERROR FOR TEST 3 IN 0 WITH RESPECT TO DIFFERENT ℓ_1-MINIMIZATION TECHNIQUES.

	Residual Error for *Test 3* vector		
	ℓ_1-minimization (8)	Lasso (9)	Elastic Net (10)
Class 1	0.63	0.95	**0.62**
Class 2	0.99	1	1
Class 3	0.99	1	1
Class 4	**0.36**	**0.57**	1

V. SPEAKER VERIFICATION EXPERIMENTS

A. Experimental Setup

The experimental set-up schematic is shown in Fig. 5. The experiments were evaluated on the condition 5 (*tel-tel*) of the NIST 2010 SRE *extended* trial set. The performance was evaluated using the equal error rate (*EER*) and a normalized minimum decision cost function (*DCF*) was calculated, where $C_{miss} = 1$, $C_{FA} = 1$ and $P_{tar} = 0.001$.

The front-end of the verification system included an energy based speech detector, which was applied to discard silence and noise frames. A Hamming window of 20ms (overlap of 10ms) was used to extract 16 MFCCs and 16 PNCCs, including C_0 and PC_0. This 16-dimensional feature vector was subjected to feature warping using a 3s sliding window, before computing delta coefficients that were appended to the MFCC/PNCC, providing a 32 dimensional feature vector.

The authors used gender-dependent UBMs of 512 Gaussians trained using NIST 2004. In the *i-vector* based systems, 400 total factors defined by the total variability matrix *T* were used. Furthermore, for channel compensation on the *i-vectors*, a LDA matrix with dimensionality reduction of 200 was trained using the Switchboard II, NIST 2004 and 2005 SRE and WCCN matrix was trained using NIST 2004 and 2005 SRE [7]. For all experiments, the results were obtained when the *i-vectors* were subjected to LDA followed by WCCN. Finally, the decision scores obtained using the cosine distance scoring were normalized using the ZT-norm. The authors used 367 female T-norm models and 274 female Z-norm utterances from NIST 2004 and 2005 SRE respectively.

B. Results

Results for the speaker verification experiments with different features and classifiers, based on the NIST 2010 SRE, are given in Table II. For comparisons across classifiers, it can be observed that SRCs were slightly inferior to the CDS back-end, in terms of EER, yet outperformed the CDS variant in terms of DCF for each of the individual features. On the other hand, in comparisons across features, the PNCC was able to achieve comparable performance to the MFCC-based systems.

To understand how the features and classifiers interact, we studied the effect of feature-classifier combinations, and evaluate the fusion of such feature-classifier subsystems as shown in Figure 5. The results for all possible pairwise combinations are given in Table III. The fusion of MFCC-CDS and PNCC-CDS provided the best performance in terms of EER, improving on an 3.99% EER baseline to 3.55%.

Furthermore, in terms of DCF, the fusion of MFCC-SRC and PNCC-SRC gave the best performance with a DCF of 0.4977. This result provides strong encouragement that sparse representation classifiers and PNCC carry complementary information to the more conventional MFCC *i-vector* cosine distance scoring system.

TABLE II
THE SPEAKER VERIFICATION RESULTS ON THE NIST 2010 SRE EXTENDED TRIALS

Features	Classifiers			
	CDS		SRC	
	EER (%)	DCF	EER (%)	DCF
MFCC	4.20	0.5963	4.56	0.5552
PNCC	**3.99**	0.6344	4.69	**0.5442**

TABLE III
FUSED SPEAKER VERIFICATION RESULTS ON THE NIST 2010 SRE EXTENDED TRIALS

Feature-Classifier	MFCC-SRC		PNCC-CDS		PNCC-SRC	
	EER (%)	DCF	EER (%)	DCF	EER (%)	DCF
MFCC-CDS	4.15	0.5535	**3.55**	0.5822	3.72	0.5226
MFCC-SRC	-	-	3.66	0.5124	3.91	**0.4977**
PNCC-CDS	-	-	-	-	3.80	0.5315

VI. CONCLUSION

This paper has outlined the PNCC-ivector-SRC speaker verification system that makes use of a PNCC front end that is mapped onto an i-vector total variability space prior to sparse representation scoring (SRC). Specifically the paper investigates the use of PNCCs in place of the more conventional MFCCs and also fusion of systems using both features. It also compares the use of the exemplar based SRC scoring to cosine distance scoring (CDS). The results included in the paper suggest strongly that the PNCC-ivector-SRC system carries complementary information to the MFCC-ivector-CDS system. Future research would look into the use of a learned dictionary in place of an exemplar based dictionary in the PNCC-ivector-SRC system.

REFERENCES

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

[2] Kim, C. and Stern, R., "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. of ICASSP*, 2012.

[3] Kua, J. M. K., Ambikairajah, E., Epps, J., and Togneri, R., "Speaker verification using sparse representation classification," in *Proc. of ICASSP*, 2011, pp. 4548-4551.

[4] Naseem, I., Togneri, R., and Bennamoun, M., "Sparse Representation for Speaker Identification," in *Proc. of ICPR*, 2010, pp. 4460-4463.

[5] Li, M., Zhang, X., Yan, Y., and Narayanan, S., "Speaker Verification using Sparse Representations on Total Variability I-Vectors," in *Proc. of INTERSPEECH*, 2011.

[6] Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P., and Dumouchel, P., "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. of INTERSPEECH*, 2009.

[7] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P., "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, pp. 788-798, 2011.

[8] Haris, B. C. and Sinha, R., "Speaker verification using sparse representation over KSVD learned dictionary," in *Proc. of National Conference on Communications (NCC)*, 2012, pp. 1-5.

[9] Epps, J. and Ambikairajah, E., "Speech Characterization and Feature Extraction for Speaker Recognition," in *Advanced Topics In Biometrics*, G. Li and H. Li, Eds., ed, ISBN 978-981-4287-84-5, pp. 45-69.

[10] Kenny, P., "Joint factor analysis of speaker and session variability: theory and algorithms," *Tech Report Online: http://www.crim.ca/perso/patrick.kenny,* 2005.

[11] Dehak, N., "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: application to speaker verification," Doctoral Dissertation, Ecole de Technologie Superieure 2009,

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

[13] Auckenthaler, R., Carey, M., and Lloyd-Thomas, H., "Score normalization for text-independent speaker verification systems," *Digital Signal Processing: A Review Journal,* vol. 10, pp. 42-54, 2000.

[14] Reynolds, D. A., "Comparison of background normalization methods for text-independent speaker verification," in *Proc. of EUROSPEECH*, 1997, pp. 963-966.

[15] Zheng, R., Zhang, S., and Xu, B., "A Comparative Study of Feature and Score Normalization for Speaker Verification," in *Advances in Biometrics*. vol. 3832, D. Zhang and A. Jain, Eds., ed: Springer Berlin / Heidelberg, 2005, ISBN 978-3-540-31111-9, pp. 531-538.

[16] Kanevsky, D., Sainath, T. N., Ramabhadran, B., and Nahamoo, D., "An analysis of sparseness and regularization in exemplar-based methods for speech classification," in *Proc. of INTERSPEECH*, 2010.