Front-end Channel Compensation using Mixture-dependent Feature Transformations for i-Vector Speaker Recognition

Taufiq Hasan and John H. L. Hansen[*]

Center for Robust Speech Systems (CRSS), Eric Jonsson School of Engineering,
University of Texas at Dallas, Richardson, Texas, U.S.A.
{Taufiq.Hasan, John.Hansen}@utdallas.edu

Abstract

State-of-the-art session variability compensation for speaker recognition are generally based on various linear statistical models of the Gaussian Mixture Model (GMM) mean super-vectors, while front-end features are only processed by standard normalization techniques. In this study, we propose a front-end channel compensation frame-work using mixture-localized linear transforms that operate before super-vector domain modeling begins. In this approach, local linear transforms are trained for each Gaussian component of a Universal Background Model (UBM), and are applied to acoustic features according to their mixture-wise probabilistic alignment, yielding an operation that is globally non-linear. We examine Principal Component Analysis (PCA), whitening, Linear Discriminant Analysis (LDA) and Nuisance Attribute Projection (NAP) as front-end feature transformations. We also propose a method, Nuisance Attribute Elimination (NAE), which is similar to NAP but performs dimensionality reduction in addition to channel compensation. We show that the proposed frame-work can be readily integrated with a standard i-Vector system by simply applying the transformations on the first order Baum-Welch statistics and transforming the UBM. Experiments performed on the telephone trials of the NIST SRE 2010 demonstrate significant performance gain from the proposed frame-work, especially using LDA as the front-end transformation.

1. Introduction

Recent advancements in session variability compensation for speaker verification is mostly due to effective application of linear statistical methods on speaker/utterance dependent GMM mean super-vectors. Methods such as, Eigenvoice [1], Eigenchannel and Joint Factor Analysis (JFA) [2], are based on the lower dimensional speaker and channel dependent subspace assumption and their variants; NAP [3] performs a linear transformation on the super-vectors aiming at projecting out nuisance directions; Total Variability (TV) modeling [4] reduces the super-vector dimension using factor analysis [1] to obtain i-Vectors, which are again processed by linear statistical techniques such as LDA, Within Class Covariance Normalization (WCCN) and Probabilistic LDA (PLDA).

Despite the success of the linear statistical models in speaker recognition and in other pattern classification tasks in general [5], acoustic features are not generally compensated using these techniques. Popular front-end channel compensation methods such as Mean and Variance Normalization (MVN) and Gaussianization [6] rely on normalizing the coefficients based on their temporal statistics alone. Linear statistical methods such as LDA and PCA have also been applied on acoustic features [7–9] in speech and speaker recognition. However, when advanced super-vector domain compensation techniques are considered, the impact many feature domain normalization techniques become insignificant [10, 11].

In this study, we propose an effective frame-work for utilizing linear statistical methods on acoustic features as a pre-processing stage, before the super-vector domain modeling begins. We first train a UBM on the development dataset and derive PCA, LDA, NAP and whitening transformation matrices for each GMM mixture. We also propose a new dimensionality reduction transformation similar to NAP, termed as nuisance attribute elimination. Conventionally, when GMMs are used for feature clustering and transformation, the most likely mixture component is obtained given the input feature vector and the corresponding transform is used [10]. This approach assumes that one feature vector aligns with a single Gaussian mixture only and new models need to be trained from the transformed feature set. In the proposed frame-work, instead to returning to the feature space and retraining the UBM, the transformations are applied to the first order Baum-Welch statistics and the UBM itself. In this way, the front-end processing can be effectively integrated into a standard i-Vector PLDA based system. Experimental evaluations show promising results using the proposed channel compensation scheme.

2. Proposed method

2.1. Mixture-wise feature transformation

Let $\mathcal{X} = \{\mathbf{x}_n | n = 1 \cdots N\}$ be the collection of all d dimensional feature vectors from the development dataset. Let us define a transformation matrix $\mathbf{A}$, and transformed feature vectors $\mathbf{z}_n$, so that,

$$\mathbf{z}_n = \mathbf{A}(\mathbf{x}_n - \mu). \tag{1}$$

Here, $\mathbf{x}_n$ represents the $d \times 1$ dimensional feature vector obtained from $\mathcal{X}$, $\mathbf{A}$ is a $d \times q$ transformation matrix where $q \leq d$, and μ is the $d \times 1$ mean vector of $\mathbf{x}_n$. The matrix $\mathbf{A}$ could be obtained from any linear statistical method, such as PCA, LDA, NAP, etc. If $q < d$, this transformation performs dimensionality reduction. Considering the variability of acoustic features in various environmental conditions and phonetic context, we presume that different regions of the feature space should have a unique transform. Thus, we utilize a UBM Λ_0 for clustering the acoustic features, given by,

$$p(\mathbf{x}_n|\Lambda_0) = \sum_{g=1}^{M} w_g \frac{\exp\left[-\frac{1}{2}(\mathbf{x}_n - \mu_g)^T \boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_n - \mu_g)\right]}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_g|^{1/2}}$$

where w_g is the mixture weight, μ_g and $\boldsymbol{\Sigma}_g$ represent the mixture mean vector and covariance matrix, and M is the number of mixtures. The transformed feature vector in the g-th mixture is:

$$\mathbf{z}_{n,g} = \mathbf{A}_g(\mathbf{x}_n - \mu_g) \tag{2}$$

[*]This project was funded by AFRL under contract FA8750-12-1-0188 (Approved for public release, distribution unlimited: 88ABW-2012-1810), and partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J.H.L. Hansen.

where $\mathbf{A}_g$ is now a mixture dependent transformation. It can be shown that $\mathbf{z}_{n,g}$ has a zero mean and a covariance matrix given by:

$$\mathbf{\Sigma}_{\mathbf{z}_g} = \mathbf{A}_g \mathbf{\Sigma}_g \mathbf{A}_g^T. \tag{3}$$

Thus, after this mixture dependent transformation is applied, the UBM Λ_0 is replaced by a transformed UBM model $\hat{\Lambda}_0$, given by,

$$p(\mathbf{z}|\hat{\Lambda}_0) = \sum_{g=1}^{M} w_g \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{z}_g}). \tag{4}$$

2.2. Integration within the i-Vector system

After feature extraction and UBM training, the first step of training a total variability matrix/i-Vector extraction is estimating the zero and first order Baum-Welch statistics. These statistics are computed from acoustic features with respect to the UBM model. For an utterance h, the zero order statistics, also known as the probabilistic count for each mixture, is extracted as,

$$N_h(g) = \sum_{n \in h} \gamma_n(g), \text{ where } \gamma_n(g) = p(g|\mathbf{x}_n, \Lambda_0). \tag{5}$$

In the proposed frame-work, the first order statistics $\mathbf{F}_h(g)$ is extracted using the transformed feature vectors in the corresponding mixtures, instead of the original acoustic features $\mathbf{x}_n$.

$$\mathbf{F}_h(g) \;=\; \sum_{n \in h} \gamma_n(g) \mathbf{z}_{g,n} = \mathbf{A}_g \sum_{n \in h} \gamma_n(g)(\mathbf{x}_n - \mu_g) \tag{6}$$

As expected, this is simply a transformed version of the centralized first order statistics [2]. Each feature vector is thus transformed according to its alignment with different mixtures that are locally effective in performing channel compensation. This process is similar to a mixture of experts model [12] for front-end channel compensation. The rest of the i-Vector system procedure follow the conventional approach, with acoustic feature dimension q and UBM model $\hat{\Lambda}_0$. Also the super-vector dimension reduces to $K = Mq$ from Md, and TV matrix size becomes $K \times R$. We define a parameter super-vector compression ratio $\alpha = K/Md$, measuring overall dimension reduction in the system.

2.3. Mixture-wise PCA (m-PCA)

Here, we describe how a mixture-wise PCA [13] is implemented in the proposed frame-work. First, a full covariance UBM Λ_0 is trained on the development data. Next, for each mixture covariance matrix $\mathbf{\Sigma}_g$, the eigenvalue decomposition is performed as:

$$\mathbf{\Sigma}_g = \mathbf{U}_g^T \mathbf{\Lambda}_g \mathbf{U}_g \tag{7}$$

where the columns of $\mathbf{U}_g$ contain the eigenvectors of $\mathbf{\Sigma}_g$, and $\mathbf{\Lambda}_g$ contains the corresponding eigenvalues in its main diagonal. Retaining the first q principal directions in the feature space within this mixture, the g-th transformation matrix is defined as:

$$\mathbf{A}_{\text{PCA-}q[g]} = \mathbf{U}_{[q]_g}^T \tag{8}$$

where $\mathbf{U}_{[q]_g}$ contains the first q columns of $\mathbf{U}_g$ corresponding to the largest eigenvalues. The transformed covariance matrix is:

$$\mathbf{\Sigma}_{\mathbf{z}_g} = \mathbf{U}_{[q]_g} \mathbf{\Sigma}_g \mathbf{U}_{[q]_g}^T = \mathbf{\Lambda}_{[q]_g} \tag{9}$$

where $\mathbf{\Lambda}_{[q]_g}$ is a $q \times q$ diagonal matrix containing the first q largest eigenvalues of $\mathbf{\Sigma}_g$. In this case, the modified UBM is given by,

$$p(\mathbf{z}|\hat{\Lambda}_{\text{PCA}}) = \sum_{g=1}^{M} w_g \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_{\mathbf{q}_g}). \tag{10}$$

Thus, using this transformation, the acoustic features are de-correlated, and when $q < d$ is set, the least important directions in the acoustic space is also suppressed.

2.4. Mixture-wise Whitening (m-WHT)

This is very similar to PCA, except that the transformation whitens the features in each mixture in addition to de-correlating them. Least important directions can be removed using this transform in the same way as PCA. The whitening transformation for the g-th mixture retaining $q \leq d$ components is given by:

$$\mathbf{A}_{\text{WHT-}q[g]} = \mathbf{\Lambda}_{[q]_g}^{-\frac{1}{2}} \mathbf{U}_{[q]_g}^T. \tag{11}$$

In this case, the new UBM $\hat{\Lambda}_{\text{WHT}}$ has an identity covariance matrix.

$$p(\mathbf{z}|\hat{\Lambda}_{\text{WHT}}) = \sum_{g=1}^{M} w_g \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{12}$$

2.5. Mixture-wise LDA (m-LDA)

To implement a mixture wise LDA transformation, we need the mixture dependent within class and between class scatter matrices, $\mathbf{S}_{\mathbf{w}g}$ and $\mathbf{S}_{\mathbf{b}g}$, respectively. We also need development data speaker labels. We proceed as follows. For each speaker $s \in S$, we compute the speaker dependent mean vector for the g-th mixture as:

$$\bar{\mathbf{x}}_{g_s} = \frac{1}{n_s} \sum_{n \in s} \gamma_n(g) \mathbf{x}_n \tag{13}$$

Here, n_s is the total number of feature frames belonging to the speaker s. From total S speakers' data, the between class and within class scatter matrices for each mixture is then computed as:

$$\mathbf{S}_{\mathbf{b}g} = \frac{1}{S} \sum_{s=1}^{S} N_s(g)(\bar{\mathbf{x}}_{g_s} - \mu_g)(\bar{\mathbf{x}}_{g_s} - \mu_g)^T \tag{14}$$

$$\mathbf{S}_{\mathbf{w}g} = \frac{1}{S} \sum_{s=1}^{S} \sum_{n \in s} \gamma_n(g)(\mathbf{x}_n - \bar{\mathbf{x}}_{g_s})(\mathbf{x}_n - \bar{\mathbf{x}}_{g_s})^T \tag{15}$$

where $N_s(g) = \sum_{n \in s} \gamma_n(g)$ is the probabilistic count of mixture g for speaker s. Next, the g-th LDA transformation matrix is computed through the following eigenvalue decomposition:

$$\mathbf{S}_{\mathbf{w}g}^{-1} \mathbf{S}_{\mathbf{b}g} = \mathbf{V}_g^T \mathbf{D}_g . \mathbf{V}_g \tag{16}$$

Here, $\mathbf{V}_g$ contains the eigenvectors as its columns and $\mathbf{D}_g$ contains the corresponding eigenvalues as its main diagonal. If $\mathbf{V}_{[q]_g}$ denotes the matrix containing $q \leq d$ columns of $\mathbf{V}_g$ corresponding to the q largest eigenvalues, the LDA transform matrix is given by:

$$\mathbf{A}_{\text{LDA-}q_g} = \mathbf{V}_{[q]_g}^T. \tag{17}$$

Using this transformation, the transformed UBM covariance matrices can be computed using (3).

One common problem in implementing multi-class LDA occurs when feature dimension becomes larger than the number of classes, leading to singular between/within-class covariance matrices [5]. For this reason, LDA is generally applied to the lower dimensional i-Vectors [4] instead of the GMM mean super-vectors. When applying LDA on acoustic features in the proposed frame-work, we observe that if the number of speakers S is larger than the acoustic feature dimension d (a condition which can be easily met) the between class covariance matrices in (14) should always be full rank. However, if in a given mixture, $N_s(g)$ is zero for a large number of speakers, $\mathbf{S}_{\mathbf{b}g}$ can be low rank. Such cases are very rare, since the same corpus will be used to train the UBM and

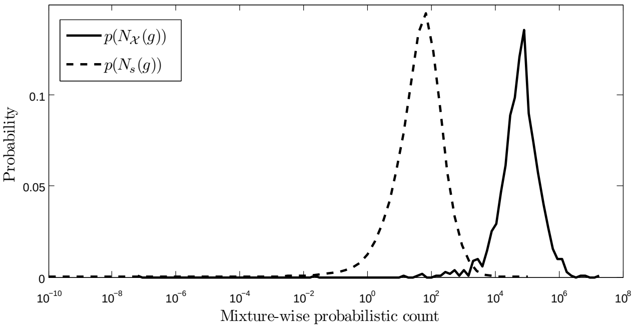

Figure 1: Distribution of mixture-wise probabilistic feature count. The distributions $p(N_{\mathcal{X}}(g))$ and $p(N_s(g))$ are obtained from 1024 mixture counts for all data, and computing the same for each 984 speakers, respectively.

estimate these matrices. Similarly, to ensure that $\mathbf{S}_{\mathbf{b}g}$ matrices in (15) are non-singular, most of the posterior probabilities for each speaker and mixture should be greater than zero.

In order to verify if these conditions are met in our system, we train a 1024 mixture UBM using 60-dimensional MFCC features extracted from our development data set. The full dataset $\mathcal{X}$ contains a total of $162,093,376$ frames obtained from 984 speakers. The probabilistic counts $N_{\mathcal{X}}(g) = \sum_{n \in \mathcal{X}} p(g|\mathbf{x}_n, \Lambda_0)$ are calculated for each mixture across the entire dataset, and $N_s(g)$ values for each speaker $s \in S$ and mixture g is computed. The probability distributions of $N_{\mathcal{X}}(g)$ and $N_s(g)$ are then estimated using normalized histograms and are shown in Fig. 1. We obtain the distributions $p(N_{\mathcal{X}}(g))$ and $p(N_s(g))$ from $M = 1024$ and $MS = 1024 \times 984$ data points, respectively. Here, we observe that for most cases $N_s(g) > 10^{-2}$ and $N_{\mathcal{X}}(g) > 10^2$. Since we have $S = 984$ speakers, $N_s(g) \sim 0$ for some mixtures for a few speakers cannot make $\mathbf{S}_{\mathbf{b}g}$ low-rank. However, if $N_{\mathcal{X}}(g)$ is close to zero for a mixture, it can lead to a singular $\mathbf{S}_{\mathbf{w}g}$ matrix. If this occurs, we do not perform the LDA transformation in that mixture and use an identity matrix instead.

2.6. Mixture-wise NAP (m-NAP)

The NAP algorithm was originally proposed in [3]. In this method, the feature space is transformed using an orthogonal projection in the channel's complementary space, which depends only on the speaker. The projection is calculated using the within-class covariance matrix. To apply NAP on acoustic features, we define a $d \times d$ projection matrix [3] of co-rank $k < d$ for the g-th mixture as:

$$\mathbf{P}_g = \mathbf{I} - \mathbf{W}_{[k]_g} \mathbf{W}_{[k]_g}^T \triangleq \mathbf{A}_{\text{NAP-}k_g} \tag{18}$$

where $\mathbf{W}_{[k]_g}$ is a rectangular matrix of low rank whose columns are the k principal eigenvectors of the matrix $\mathbf{S}_{\mathbf{w}g}$ in (15). The transformed UBM covariance matrices are found using (3):

$$\mathbf{\Sigma}_{\mathbf{z}_g} \quad = \quad \mathbf{P}_g \mathbf{\Sigma}_g \mathbf{P}_g^T. \tag{19}$$

Since NAP removes some nuisance directions from the feature space in each mixture, the operation in (19) on the mixture covariance matrices $\mathbf{\Sigma}_g$, results in rank-deficient, and thus non-invertible transformed matrices $\mathbf{\Sigma}_{\mathbf{z}_g}$. To avoid inverting $\mathbf{\Sigma}_{\mathbf{z}_g}$, we use its pseudo-inverse in our system, which is calculated using the Singular Value Decomposition (SVD) method presented in [14]. We note that, NAP does not reduce the feature dimension. Thus, the super-vector compression ratio $\alpha = 1$ in this case.

2.7. Mixture-wise Nuisance Attribute Elimination (m-NAE)

We propose a dimensionality reduction transformation that uses the same principles as NAP, but eliminates the nuisance directions from the feature space instead of projecting them out. In this way, the transformed UBM covariance matrices are smaller in size, but still full rank and invertible. For the proposed method, we transform the features by the first $q = (d - k)$ eigenvectors corresponding to the largest eigenvalues of $\mathbf{S}_{\mathbf{w}g}$ denoted by $\mathbf{W}_{[q]_g}$. Here, k is the number of dimensions eliminated. The NAE transform is given by:

$$\mathbf{A}_{\text{NAE-}q_g} = \mathbf{W}_{[q]_g}^T. \tag{20}$$

Here, the acoustic features are dimensionality reduced from d to q.

3. Experiments

We perform our experiments on the male trials of NIST SRE 2010 telephone train/test condition (condition 5, normal vocal effort). A standard i-Vector system with a Gaussian Probabilistic Linear Discriminant Analysis (PLDA) [15] back-end is used for the evaluation. Different blocks of the system is described below.

3.1. Feature Extraction

For voice activity detection (VAD), a phoneme recognizer [16] and energy based scheme is used. A 60-dimensional feature vector (19 MFCC +Energy + Δ + $\Delta\Delta$) is extracted, using a 25 ms analysis window with 10 ms shift and filtered by feature warping using a 3-s sliding window [6].

3.2. UBM Training

A gender dependent full-covariance UBM with 1024 mixtures is trained on utterances selected from Switchboard II Phase 2 and 3, Switchboard Cellular Part 1 and 2, and the NIST 2004, 2005, 2006 SRE enrollment data. For training, we used the HTK toolkit using 15 iterations per mixture split.

3.3. Total variability modeling

For the total variability matrix training, the UBM training dataset is utilized. i-Vector dimension was set to 400. All i-Vectors are first whitened and then length normalized [15].

3.4. Back-end channel compensation and scoring

A Gaussian PLDA with full-covariance noise model is used for both session variability compensation and scoring [15] . In this model, the only free parameter is the number of Eigenvoices N_{EV}, which was set to 150.

4. Results

The results of our experiments are summarized in Table 1. The "Baseline" system refers to the i-Vector PLDA system. For the proposed front-end channel compensation methods m-PCA, m-WHT, m-LDA, m-NAP and m-NAE, various parameter values shown in Table 1 are used. The performance metrics used are: Equal Error Rate (%EER), and minimum Detection Cost Functions of NIST SRE 2008 (DCF$_{\text{old}}$) and 2010 (DCF$_{\text{new}}$). From the results, we observe that m-PCA and m-WHT can generally improve the system performance upto $\sim$ 10% relative to the baseline. Improvements are observed for both with and without dimensionality reduction. The m-LDA method provides the best performance of all the transforms. An EER of 1.718% is attained yielding 19.4% relative improvement compared to the baseline system when $q = 40$ is used. The techniques m-NAP and m-NAE performed worse compared to m-PCA, m-WHT and m-LDA, with the proposed m-NAE technique generally outperforming m-NAP. Given the simplicity of the transforms used, the performance gains clearly demonstrate the effectiveness of the proposed channel compensation scheme.

Table 1: Comparison between baseline i-Vector and proposed systems with respect to %EER, DCF$_{\text{old}}$ and DCF$_{\text{new}}$ for $N_{EV} = 150$. Percent relative improvement (%r) and super-vector compression ratio (α) are also shown.

System		α	**%EER/%r**	**DCF$_{\text{old}}$/%r**	**DCF$_{\text{new}}$/%r**
Baseline		1.00	2.13284	0.11308	0.39845
Method	*Parameter*				
m-PCA	$q = 42$	0.70	1.827/14.35	0.106/6.45	0.397/0.32
	$q = 48$	0.80	2.030/4.82	0.108/4.56	**0.363/8.92**
	$q = 60$	1.00	1.899/10.98	0.105/7.40	0.387/2.90
m-WHT	$q = 42$	0.70	1.887/11.55	0.105/7.29	0.379/4.79
	$q = 48$	0.80	1.908/10.54	0.109/3.56	0.372/6.60
	$q = 60$	0.80	1.920/9.96	0.108/4.79	0.381/4.38
m-LDA	$q = 36$	0.60	2.065/3.20	0.105/6.86	0.384/3.71
	$q = 40$	0.66	**1.718/19.43**	**0.096/15.05**	0.40/-0.47
	$q = 48$	0.80	1.857/12.94	0.107/5.70	0.389/2.49
m-NAP	$k = 5$	1.00	2.011/5.71	0.113/0.22	0.411/-3.2
	$k = 10$	1.00	2.130/0.11	0.115/-1.3	0.413/-3.7
	$k = 20$	1.00	2.108/1.16	0.117/-3.7	0.44/-10.3
m-NAE	$k = 5$	0.92	1.982/7.05	0.112/0.79	0.416/-4.4
	$k = 10$	0.83	2.079/2.54	0.106/6.55	0.418/-4.8
	$k = 20$	0.66	2.120/0.61	0.122/-7.8	0.45/-11.6

Table 2: Linear score fusion of baseline and proposed systems

	Individual system performances			
	System	**%EER**	**DCF$_{\text{old}}$**	**DCF$_{\text{new}}$**
(i)	Baseline	2.13284	0.11308	0.39845
(ii)	m-PCA$_{42}$-i-Vector	1.82672	0.10579	0.39719
(iii)	m-WHT$_{42}$-i-Vector	1.88659	0.10484	0.37935
(iv)	m-LDA$_{40}$-i-Vector	1.71848	0.09606	0.40034
	Fusion system performances			
1	Fusion of (i) & (ii)	1.81949	0.09845	0.35695
2	Fusion of (i) & (iii)	1.77720	0.09817	0.36436
3	Fusion of (i) - (iv)	**1.68627**	**0.09307**	**0.35549**

In Table 2, fusion performance of the following four systems are presented: (i) Baseline, (ii) m-PCA$_{42}$, (iii) m-WHT$_{42}$, and (iv) m-LDA$_{40}$. From these results, it is clear that proposed systems provide complimentary information compared to the baseline system. The best performance is attained by fusing all four of these systems to reach a performance of, EER = 1.686%, DCF$_{\text{old}}$ = 0.093 and DCF$_{\text{new}}$ = 0.355. In Fig. 2, the performance comparison of these systems are shown using a Detection Error Tradeoff (DET) curve.

5. Conclusions

In this study, we have proposed a front-end channel compensation frame-work utilizing various linear statistical methods operating in each mixture of a UBM. Mixture-localized formulations of PCA, LDA, whitening and NAP were described in the proposed frame-work. A new transformation termed nuisance attribute elimination was also presented. Instead of regenerating the acoustic features, the mixture-localized transforms were applied to the UBM and the first-order Baum-Welch statistics, and thus, were integrated within a standard i-Vector PLDA speaker recognition system. Experiments were performed on NIST SRE 2010 telephone trials demonstrating the effectiveness of the proposed channel compensation frame-work. Significant performance improvements compared to the baseline system were obtained when using LDA as a front-end transformation.

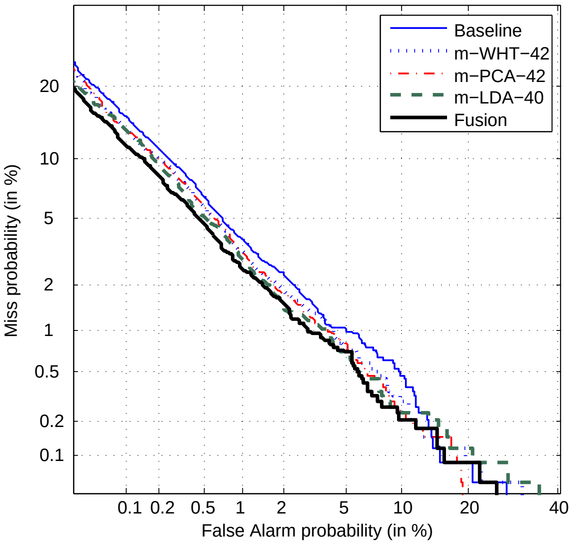

Figure 2: Performance comparison between proposed, baseline and fusion systems demonstrated using Detection Error Trade-off (DET) curves.

6. References

[1] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech and Audio Process.*, vol. 13, no. 3, pp. 345–354, May 2005.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus Eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[3] A. Solomonoff, W. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proc. ICASSP*, vol. 1, 2005, pp. 629–632.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 99, pp. 788 – 798, May 2010.

[5] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *Intl. J. of Imaging Sys. and Tech.*, vol. 15, no. 5, pp. 252–260, 2005.

[6] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey*, 2001, pp. 213–218.

[7] K. Y. Lee, "Local fuzzy PCA based GMM with dimension reduction on speaker identification," *Pattern Recogn. Lett.*, vol. 25, pp. 1811–1817, December 2004.

[8] T. Eisele, R. Haeb-Umbach, and D. Langmann, "A comparative study of linear feature transformation techniques for automatic speech recognition," in *Proc. ICSLP*, vol. 1, 1996, pp. 252–255.

[9] Q. Jin and A. Waibel, "Application of LDA to speaker recognition," in *Proc. ICSLP*, 2000, pp. 250–253.

[10] L. Burget *et. al.*, "Analysis of feature extraction and channel compensation in a GMM speaker recognition system," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15, no. 7, pp. 1979–1986, Sept. 2007.

[11] M. Alam, P. Ouellet, P. Kenny, and D. O'Shaughnessy, "Comparative evaluation of feature normalization techniques for speaker verification," *Advances in Nonlinear Speech Process.*, pp. 246–253, 2011.

[12] M. Jordan and R. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[13] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural computation*, vol. 11, no. 2, pp. 443–482, 1999.

[14] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, pp. 205–224, 1965.

[15] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-Vector length normalization in speaker recognition systems," in *Proc. Interspeech*, Florence, Italy, Oct. 2011, pp. 249 – 252.

[16] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, vol. 1, May 2006.