Chapter 5
Robust Speaker Modeling for Speaker Verification in Noisy Environments

Abstract The present chapter explores robust speaker modeling methods for speaker verification in noisy environment. The focus is specifically laid on building hybrid classifiers based on the combination of generative and discriminative models (e.g., Gaussian Mixture Models (GMMs) and Support Vector Machines (SVMs)). For improving the performance of the proposed speaker verification systems, utterance partitioning methods are used. The discussion is closely followed by state-of-the-art variants of GMM supervector based approaches (i.e., i-vectors) and algorithms for combining robust classifiers.

The application of stochastic feature compensation for speaker verification (SV) as studied in Chap. 4, is associated with certain drawbacks. Firstly, it depends on the availability of stereo data which is expensive to acquire. Secondly, a priori knowledge about a speaker's test environment is assumed i.e., the background environment during evaluation should be reflected in the stereo training data. Lastly, substantial amount of data may be required for the joint probability modeling techniques. However, in real-life scenarios the test environments are often unknown and time-varying (non-stationary). SV applications deployed in hand-held devices are additionally expected to perform in real-time with minimal data requirements.

As an alternative strategy, model compensation and robust speaker modeling methods can be explored. The role of these two methods have been briefly explained in Chaps. 1 and 2, respectively. We had also emphasized on certain limitations of the conventional model compensation methods such as requirement of clean speaker models, dependence on a mathematical representation of the noise corruption process. Additionally, popular model compensation methods like Parallel Model Combination demand substantial amount of training data and high computational resources which may not be frequently available.

The present chapter explores robust speaker modeling methods for SV in noisy environment. The focus is specifically laid on building hybrid classifiers based on the combination of generative and discriminative models (e.g., Gaussian Mixture

K.S. Rao and S. Sarkar, *Robust Speaker Recognition in Noisy Environments*,
SpringerBriefs in Electrical and Computer Engineering,
DOI 10.1007/978-3-319-07130-5_5, © The Author(s) 2014

Models (GMMs) and Support Vector Machines (SVMs)). The discussion is closely followed by state-of-the-art variants of GMM supervector based approaches (i.e., i-vectors) and algorithms for combining robust classifiers.

5.1 GMM-SVM Combined Approach for Speaker Verification

The traditional GMM-UBM based speaker verification system requires a Universal Background Model (UBM) [1] and a *Maximum aPosteriori* (MAP) adapted Gaussian Mixture Model (GMM) to represent the impostor and actual speaker classes, respectively. During the evaluation stage, a test utterance is classified based on its statistical similarities with the claimed target speaker model (GMM) and the background model (UBM). Gaussian Mixture Models (GMMs) are extensively applied for speaker modeling due to their strong probabilistic framework, scalability to large training sets and high recognition accuracy. GMMs belong to the family of generative models in which each speaker is modeled individually. Performance accuracy of a SV system is usually increased when these generative models are brought into a discriminative framework using Support Vector Machines (SVMs) [2].

SVMs have been established as an effective discriminative classifiers for speaker recognition tasks [3]. Through a non-linear function (i.e., kernel) a SVM maps input vectors to a high dimensional space where classes are more likely to be linearly separable [4]. However, fixed length representation of utterance is crucial for SVM training in order to avoid large target models and slow scoring. This had initially led to concept of ‘sequence kernels’[5] where variable length utterances were mapped to fixed length vectors. A robust representation was proposed later in which fixed size ‘supervectors’ constructed by stacking the means of MAP adapted GMMs were used as an input to SVM kernels [2]. Conventionally, a GMM based system calculates log-likelihood probabilities (scores) of features extracted at a frame level. In contrast, supervectors provide numerical comparison of speech utterances as an entire sequence rather than frame-wise probabilities thus preserving information which can be otherwise discarded in the frame-level [5]. Supervectors are attractive due to a number of reasons. Besides providing a high-dimensional representation for SVM classification, supervectors can distinctly characterize speaker and channel information [6]. Additionally, they can be used to compensate for channel and session variabilities [7]. In this chapter we shall explore the robustness of supervector based speaker modeling approaches for SV in noisy environment. In the following sections we briefly introduce SVMs and describe the process of integrating GMM supervectors in the SVM framework. Figure 5.1 shows the various stages of a GMM-SVM based SV system, each of which are elaborated in the remaining part of the section.

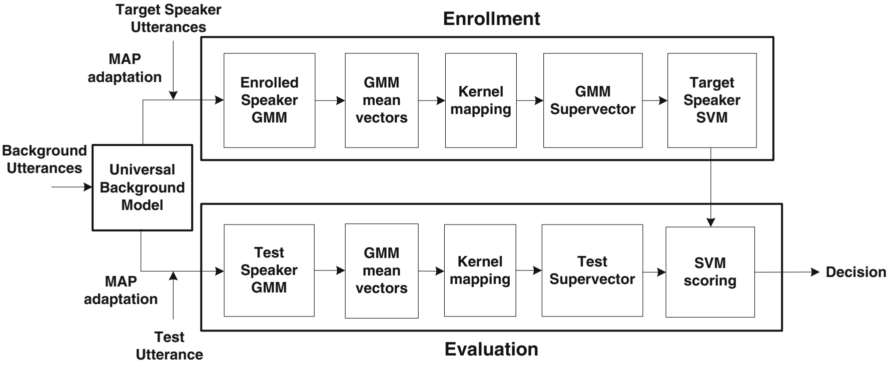

Fig. 5.1 Block diagram of the GMM-SVM framework for speaker verification

5.1.1 Support Vector Machines

A support vector machine (SVM) is a binary classifier [4]. Using labeled training vectors, a SVM optimizer finds a decision hyperplane that maximizes the margin of separation between two classes (target speaker and impostor). The classifier equation is given as follows:

$$y(x) = \sum_{i=1}^{L} \alpha_i t_i K(x_i, x) + d \tag{5.1}$$

where $\alpha_i > 0$ are the Lagrange multipliers and x_i are the Support vectors. Both these parameters are learned during the optimization process. $t_i \in \{-1, +1\}$ are the training labels, K is the desired kernel mapping and d is a bias parameter. For any input vector x the actual output $y(x)$ is compared with a decision threshold for final classification. The kernel function is constrained to satisfy the Mercer's conditions [4], so that they can be expressed as

$$K(x, y) = S(x)^T S(y) \tag{5.2}$$

where $S(x)$, $S(y)$ are high dimensional mappings for inputs x and y, respectively.

5.1.2 Construction of GMM Supervectors

The GMM-UBM framework for SV was discussed in Chap. 3. During enrollment, the pre-estimated parameters of the UBM (i.e., mean and covariance (optionally)) are modified by MAP adaptation using a target speaker's utterance to produce speaker specific GMMs given by the following equation

$$p(x) = \sum_{i=1}^{M} w_i \mathscr{N}(x; m_i, \Sigma_i) \tag{5.3}$$

where m_i, Σ_i denotes the mean and covariance of the ith multivariate Gaussian component $\mathscr{N}()$ and M is the total number of GMM components. The high-dimensional vector obtained by concatenating the mean vectors m_i of each Gaussian is generally termed as a *supervector*. Therefore D dimensional feature vectors in the input space are converted to a single $M \times D$ dimensional supervector irrespective of the number of feature vectors available. In other words, this process transforms variable length utterances to a unique fixed-size vector which carries speaker information. This representation is in conformity with Eq. (5.2) where two arbitrary utterances a and b from the input space can be compared in the supervector space using the relation $K(a, b) = S(a)^T S(b)$, where K is the kernel function and $S(a)$, $S(b)$ are the supervectors obtained from utterances a and b, respectively. The supervector construction process can be summarized as follows [8]

1. A target speaker GMM is obtained by MAP adaptation of the UBM using the speaker's enrollment utterance.
2. A kernel function is used to transform parameters of each GMM component to a fixed length vector. The vector corresponding to the ith GMM component constitutes the ith subvector of a supervector.
3. All the subvectors are concatenated to obtain a high-dimensional supervector.

5.1.3 SVM Kernels

The main design component in an SVM is the kernel, which is an inner product in the SVM feature space. The basic goal in SVM kernel design is to find an appropriate metric in the SVM feature space relevant to the classification problem. In this section we define the kernels used in our work.

5.1.3.1 KL Divergence Kernel

The Kullback Leibler divergence (KL div) is a non-symmetric distance measure between two probability distributions. Given two distributions p_a and p_b, the KL divergence between them is defined as

$$D_{KL}(p_a, p_b) = \int p_a(x) \log\left(\frac{p_a(x)}{p_b(x)}\right) \mathrm{d}x \tag{5.4}$$

However the KL divergence doesn't satisfy the Mercer's condition for a valid kernel. As a solution a symmetrized version of the KL divergence, obtained by bounding the expression by log-sum inequality was proposed as a kernel in [2]. The final version was a linear function of two MAP adapted GMMs p_a and p_b corresponding to utterances a and b. Ignoring adaptation of the UBM covariance matrix Σ_i^u and weights w_i, the resulting Kernel is given by

$$K_{KL}(p_a, p_b) = \sum_{i=1}^{M} (\sqrt{w_i}(\Sigma_i^u)^{-1/2} m_i^a)^T (\sqrt{w_i}(\Sigma_i^u)^{-1/2} m_i^b) \tag{5.5}$$

where m_i^a and m_i^b are the ith component means of p_a and p_b respectively. Thus the ith subvector of the GMM supervector for any utterance λ is given by

$$s_i^\lambda = \sqrt{w_i}(\Sigma_i^u)^{-1/2} m_i^\lambda \qquad i = 1, 2, \ldots, M$$

The final supervector obtained by concatenating the subvectors is given by $S^\lambda = [s_1^T, s_2^T, \ldots s_M^T]^T$.

5.1.3.2 GMM-UBM Mean Interval Kernel

The Bhattacharya distance between two probability distribution p_a and p_b is given by

$$D_{Bhatt}(p_a, p_b) = \int \sqrt{p_a(x) p_b(x)} \mathrm{d}x \tag{5.6}$$

For multivariate Gaussian distributions, computing this measure requires estimation of the covariance matrices which in turn demands a high amount of training data. Hence this measure is avoided in practical scenarios. However, it was shown in [9] that second order statistics derived from limited amount of training data could provide supplementary discriminative information, when used effectively. The GMM-UBM Mean Interval (GUMI) kernel based on the Bhattacharya distance between GMMs p_a and p_b, as proposed in [9] is given by

$$K_{GUMI}(p_a, p_b) = \sum_{i=1}^{M} (m_i^b - m_i^a)^T \left[\frac{\Sigma_i^a + \Sigma_i^b}{2}\right]^{-1} (m_i^b - m_i^a) \tag{5.7}$$

Considering the statistical similarities of a adapted speaker GMM and the UBM the ith subvector of the GMM supervector for an utterance λ is given by

$$s_i^\lambda = \left[\frac{\Sigma_i^\lambda + \Sigma_i^u}{2}\right]^{-1/2} (m_i^\lambda - m_i^u) \qquad i = 1, 2, \ldots, M$$

The final supervector obtained by concatenating the subvectors is given by $S^\lambda = [s_1^T, s_2^T, \ldots s_M^T]^T$.

5.1.4 SVM Scoring

Given a supervector S^{test} derived from a test utterance X^{test} the Kernel scoring is obtained as follows:

$$Score(X^{test}) = \sum_{i=1}^{L} \alpha_i t_i K(X^i, X^{test}) + d = \left(\sum_{i=1}^{L} \alpha_i t_i S^i\right)^T S^{test} + d \qquad (5.8)$$

where X^i are the sequence of learned support vectors, S^i are the supervectors corresponding to X^i, α_i are the non-zero Lagrange multipliers and $t_i \in \{-1, +1\}$ depending on the class of vector X^i. L is the total number of support vectors and d is a bias term. K is either of the two kernels used and T denotes matrix transpose.

5.1.5 Experimental Setup

All experiments are conducted on the NIST-SRE-2003 database. The data consists of single training utterances of approximately 2 min length from each of 356 enrolled speakers and 3,500 test utterances (approximately 10–15 s each) for evaluation. The stages involved in developing the GMM-SVM based SV system are briefly discussed in the following sections.

5.1.5.1 Background Simulation and Feature Extraction

The background simulation and feature extraction process has already been discussed in Chaps. 3 and 4. Summarily, all training and test utterance were degraded with additive noises (car, factory, pink and white) collected from the NOISEX-92 database. Two types of background simulations were carried out viz., (i) uniform backgrounds in which an entire utterance (training/testing) was degraded with a particular type of noise at 0, 5 and 10 dB SNRs and (ii) varying backgrounds in which non-overlapping segments of an utterance (training/testing) were individually degraded with a specific type of noise at 0, 5, 7 and 10 dB SNRs. After an energy-based voiced activity detection, 39-dimensional feature vectors (consisting of 13 MFCCs + Δ + $\Delta\Delta$ excluding C_0) derived from a 26 channel mel-scaled filterbank, were extracted from pre-emphasized speech frames of 20 ms with a frame-overlap of 10 ms. All feature vectors were subjected to cepstral mean subtraction followed by cepstral variance normalization.

5.1.5.2 Speaker Modeling

A 1,024-component GMM constructed from 20 h of speech (10 h male + 10 h female) collected from the SwitchBoard II corpus, was used as the UBM. Three hundred and fifty-six target speaker GMMs were obtained by MAP-adaptation of the UBM using the noise-degraded enrollment utterances in each dataset described in Sect. 5.1.5.1. A GMM supervector was constructed from each target speaker GMM as described in Sect. 5.1.2. The kernels described in Sect. 5.1.3 were individually used for mapping. The supervectors obtained were of 39,936 dimension (1,024 mixtures × 39 dim mean). For discriminative modeling each target speaker in a dataset was distinguished from the remaining 355 background speakers (impostors). A SVM for each speaker was trained with the speaker's supervector labelled as $+1$ and the background supervectors labelled as -1, respectively. The KL divergence and GMM-UBM mean interval kernels were used for SVM training as described in Sect. 5.1.1.

5.1.6 Performance Evaluation

All experiments were performed in matched condition i.e., training and evaluation phases having similar backgrounds. An additional evaluation was performed in clean condition. The 3,500 test utterances in each noise-corrupted dataset were transformed to supervectors prior to SVM scoring (Eq. 5.8). The NIST-2003 primary task was carried out in which each noisy test utterance (supervector) was evaluated against 11 target speaker models (SVMs) from the same dataset. The equal error rate (EER) and minimum DCF (MinDCF) values were used as metrics for performance evaluation. The standard GMM-UBM based SV systems have been used as a baseline system for performance comparison.

Table 5.1 summarizes the performance of the various SV systems developed in uniform noisy environments. The improvement in performance accuracy is clearly apparent in case of the GMM-SVM based systems in comparison to the baseline. This is manifested by a consistent reduction in EER and MinDCF values across all 12 types of noisy environments. The performance accuracy is observed to degrade non-uniformly with decreasing SNR levels. The loss in accuracy of the GMM-SVM based systems with increasing noise distortion, is correlated with that of the baseline system. An average increment of 4.48, 3.69 and 3.93 % EER values is observed for a transition from 10 to 5 dB SNR in case of the baseline, GMM-SVM (KL div) and GMM-SVM (GUMI), respectively across all four backgrounds. The same observation sequence for the 5 to 0 dB SNR transition shows average increments of 2.12, 1.07 and 0.98 % EERs which indicates that the SVM based systems are relatively more robust towards noise degradations. However the averaged metric values does not characterize individual noise behavior. For instance we observe fractional performance improvement in case of factory noise at 0 and 5 dB for GMM-SVM (KL div). A general order of precedence (best to worst) of the noisy

Table 5.1 Performance of the SV systems in uniform background environments and clean conditions

		GMM-UBM		GMM-SVM (KL div)		GMM-SVM (GUMI)	
SNR	Noises	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0 dB	Car	18.04	0.071	14.32	0.063	13.82	0.058
	Factory	23.17	0.089	21.27	0.086	20.32	0.086
	Pink	26.65	0.097	21.23	0.087	19.92	0.085
	White	30.98	0.097	22.72	0.076	22.67	0.079
5 dB	Car	18.11	0.071	13.77	0.059	13.10	0.058
	Factory	20.96	0.085	20.87	0.085	19.92	0.084
	Pink	23.89	0.092	19.65	0.086	19.06	0.085
	White	27.41	0.094	20.96	0.074	20.73	0.072
10 dB	Car	15.44	0.068	12.24	0.047	11.38	0.046
	Factory	16.44	0.072	14.81	0.053	14.50	0.052
	Pink	18.65	0.081	15.67	0.058	15.72	0.057
	White	21.91	0.087	17.75	0.065	15.49	0.067
Clean		06.93	0.033	06.72	0.030	06.44	0.030

backgrounds is noticed in terms of overall performance of the GMM-SVM based systems. Ignoring minor exceptions in case of 0 and 10 dB SNRs the order is car, pink, factory and white. This is in contrast with the baseline where the performance in pink noisy background is worse than that of factory background for all SNR levels. A comparison amongst the GMM-SVM based systems reveals that the SVMs with GUMI kernel performs moderately better than those with KL div kernel with an average reduction of 0.72 % EER across all environments.

Figure 5.2 shows the DET plots for the SV systems in (a) Car (b) Factory (c) Pink and (d) White noisy backgrounds at various SNRs. The DET curves of the GMM-UBM and GMM-SVM based systems are denoted by a set of black, red (GUMI) and blue (KL div) lines, respectively. The red and blue lines show a shift towards the origin indicating joint reduction of error probabilities. Additionally, a distinct anticlock-wise rotation in the red and blue set of curves can be noticed in comparison to the black curves (baseline) which is particularly prominent in case of factory, pink and white noise. This characteristic suggests higher reduction in 'miss' error rates compared to the 'false alarm' rates which is also evident from significant reduction in MinDCF values. Table 5.2 shows the performance improvement of the GMM-SVM based systems compared to the baseline in terms of the 'Relative Equal Error Rate' (EER_R) defined as $EER_R = \frac{EER_B - EER_V}{EER_B} \times 100\,\%$ where EER_B and EER_V are the EER values of the baseline (GMM-UBM) and GMM-SVM based systems, respectively. The SV systems with KL div kernels score average EER_R values of 21.17, 6.18, 18.02 and 23.06 % for car, factory, pink and white noisy backgrounds, respectively. The GUMI kernel based SV systems perform even better with average EER_R values of 25.78, 9.69, 20.39 and 26.83 % in the same backgrounds. Figure 5.3 shows the changes in (a) EER and (b) Relative EER of the SV systems at different SNRs in uniform noisy environments.

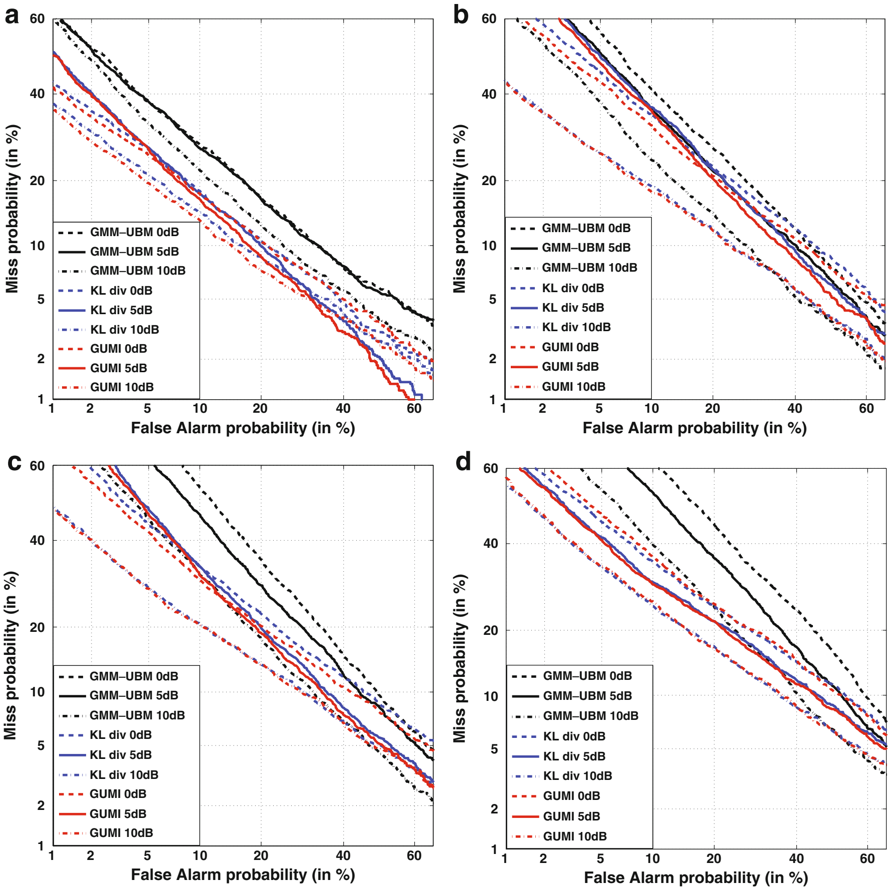

Fig. 5.2 DET plots of the SV systems in uniform background environment with (**a**) car (**b**) factory (**c**) pink and (**d**) white noise. The *black*, *blue* and *red colors* indicate GMM-UBM, GMM-SVM trained using KL div kernel and GUMI kernel, respectively

The EER values reduce consistently with increasing SNRs. However, the individual EER_R values across each SNR shows distinct behavior for each noise. In most cases there is an abrupt change at the 5 dB SNR level with the exception of pink noise which shows a consistent linear reduction for both types of SVMs.

Table 5.3 summarizes the performance of the SV systems developed in varying background environments. Though a direct comparison is inappropriate, an overall inferior performance is observed in contrast to SV systems in uniform noisy backgrounds. The utterances used for training these systems had short segments corrupted with the noises individually used for uniform background simulation, at a fixed SNR (see Chap. 3). Thus the average SV performance across all uniform backgrounds at a fixed SNR was compared with the SV performance in varying

Table 5.2 Relative equal error rates for GMM-SVM based SV systems in uniform background environments

	Relative equal error rate EER_R (%)					
	SNR (0 dB)		SNR (5 dB)		SNR (10 dB)	
Noises	KL div	GUMI	KL div	GUMI	KL div	GUMI
Car	20.62	23.39	23.96	27.66	20.73	26.30
Factory	08.20	12.30	00.43	04.96	09.91	11.80
Pink	20.38	25.25	17.75	20.22	15.98	15.71
White	26.66	26.82	23.53	27.41	18.99	29.30

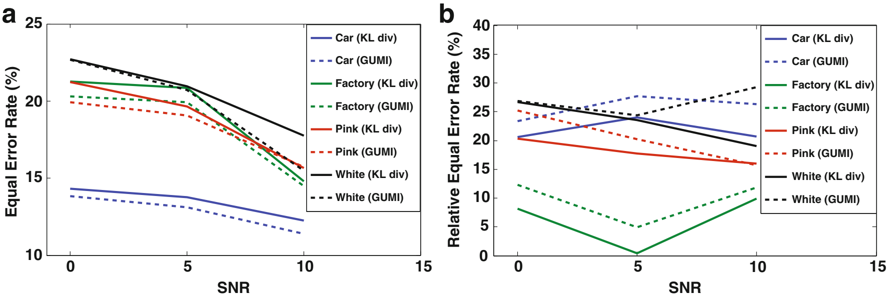

Fig. 5.3 (**a**) Equal error rates and (**b**) relative equal error rates of GMM-SVM based SV systems at different SNRs in uniform background environments

Table 5.3 Performance of the SV systems in varying background environments

SNR	GMM-UBM		GMM-SVM (KL div)		GMM-SVM (GUMI)	
(dB)	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0	27.05	0.094	23.48	0.086	22.76	0.085
5	25.74	0.086	22.18	0.080	21.32	0.081
7	25.29	0.083	19.74	0.073	19.11	0.071
10	21.86	0.080	18.65	0.072	16.44	0.069

background at the same SNR. The average EER values of the baseline systems across uniform backgrounds obtained earlier (see Table 5.1) were 24.70, 22.59 and 18.11 % for 0, 5 and 10 dB SNRs, respectively. Similarly, average EER values for the GMM-SVM (KL div) and GMM-SVM (GUMI) systems at the three SNR levels were 19.89, 18.81, 15.12 % and 19.18, 18.20, 14.28 %, respectively. In contrast, performance of the baseline systems in varying backgrounds shows an average EER increment of 3.08 %, ignoring the 7 dB SNR value. A likewise comparison with the corresponding SVM based systems with KL div and GUMI kernels, shows average increments of 3.50 and 2.95 % EERs, respectively. A possible explanation to this behavior is the inadequate amount of data used for capturing the statistics of non-stationary noise. The rapid change in noise could also causes a greater

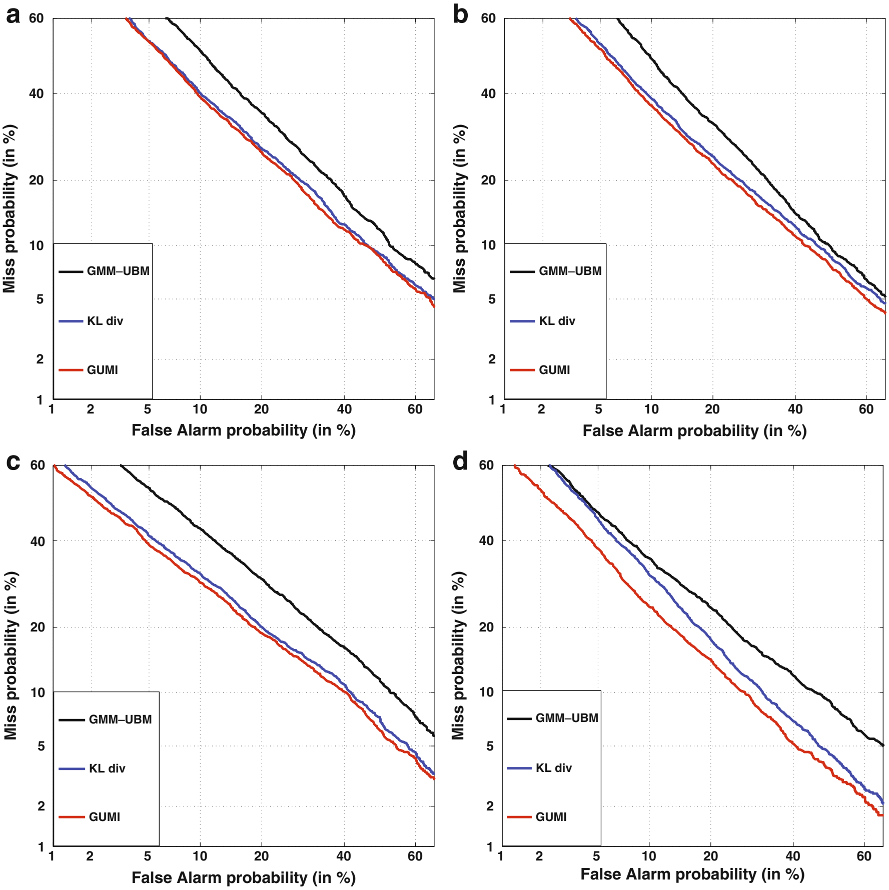

Fig. 5.4 DET plots for the SV systems in varying background environments at (**a**) 0 dB (**b**) 5 dB (**c**) 7 dB and (**d**) 10 dB SNRs. The *black*, *blue* and *red colors* indicate GMM-UBM, GMM-SVM trained using KL div kernel and GUMI kernel, respectively

degree of mismatch during the evaluation phase. The effect is more prominent in case of the baseline systems and comparatively less for the others. However, certain changes in the behavior of the GMM-SVM based systems are apparent. Unlike the uniform background case, the use of costly covariance kernels (GUMI) provides a better improvement of 1.11 % EER over the KL div kernels, when averaged across all SNRs.

Figure 5.4 shows the DET plots of the SV systems developed in varying background environment at (a) 0 dB (b) 5 dB (c) 7 dB and (d) 10 dB SNRs. The characteristics of the blue (KL div) and red (GUMI) curves are in contrast to those in Fig. 5.2. In most cases, there are no apparent rotation in the curves though an overall shift towards the origin can be noticed. In fact, the red and blue lines shows a slight rotation in clock-wise direction for 10 dB SNR despite preserving a notable difference in false alarm rates with respect to the baseline. Interestingly, the set of red and blue lines show similar properties in terms of the slope, shape and alignment with each other. The overall improvement in average MinDCF values are 8×10^{-3} and 9.25×10^{-3} for the SVM based systems with KL div and GUMI kernels, respectively. This is significantly lower in comparison to the uniform background scenarios. The overall inferior performance of the SV systems in varying background environment encouraged the use of a SVM-based channel compensation method prior to SVM training.

5.1.6.1 Nuisance Attribute Projection

Nuisance Attribute Projection (NAP) [10] is a commonly applied session compensation technique for GMM-SVM based SV systems. NAP aims to remove components (nuisance attributes) from the supervector space which are irrelevant for speaker recognition and may carry information related to channel, background etc. In other words, it eliminates the subspace which causes variabilities. This is achieved by an orthogonal projection of the supervectors in the channel's complementary space. A projection matrix P is trained using an auxillary set of speakers carrying various channel information as given by

$$P = I - vv^T \tag{5.9}$$

where v is a low rank rectangular matrix whose columns are given by 'k' eigenvectors with highest eigenvalues of the supervector's within-class covariance matrix. Thus a new linear kernel is constructed for inputs x and y after NAP operation on the supervectors $S(x)$ and $S(y)$ which is given by

$$K(x, y) = [PS(x)]^T [PS(y)] \tag{5.10}$$

The formal steps of calculating the NAP projection matrix are given as follows

1. A set of supervectors is constructed from a target speaker's enrollment utterances.
2. For each speaker, the mean of the supervectors is subtracted from each supervector in the set to subdue intra-speaker variability.
3. A large matrix V is formed whose columns constitute mean-removed supervectors from all speakers. This matrix is expected to contain session information.
4. The within class covariance matrix W of matrix V is calculated as $W = VV^T$ and subjected to eigen decomposition.
5. The eigenvectors having the largest 'k' eigenvalues are used to form the rectangular matrix v. The integer 'k' (called NAP rank), is usually determined empirically.
6. The projection matrix P is calculated by Eq. (5.9).

Table 5.4 Comparison of the performances of the GMM-SVM based SV systems in varying background environments with and without NAP compensation

	GMM-SVM				GMM-SVM + NAP			
SNR	KL div		GUMI		KL div		GUMI	
(dB)	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0	23.48	0.086	22.76	0.085	22.22	0.084	21.27	0.083
5	22.18	0.080	21.32	0.081	21.05	0.079	20.14	0.080
7	19.74	0.073	19.11	0.071	18.74	0.072	18.06	0.070
10	18.65	0.072	16.44	0.069	17.75	0.070	15.77	0.067

The NAP matrix was trained using 400 utterances collected from a set of 100 speakers of the NIST-SRE-2004 corpus. Steps 1–6 define the ideal method for estimating the NAP matrix. However, a direct application of Step 4 was infeasible due to the large size of supervectors (i.e., 39,936). As an alternative strategy, an eigenvector matrix v' was first constructed by eigen decomposition of the matrix $W' = \frac{1}{N}V^TV$ where N is the number of supervectors. The required matrix v was then obtained by the operation $v = N^{-1/2}Vv'\lambda^{-1/2}$ where λ is a diagonal matrix containing eigenvalues of the matrix W'. NAP transformation produced four new sets of supervectors (one for each SNR), which were subjected to SVM training and evaluation as explained earlier in Sects. 5.1.5 and 5.1.6, respectively. A NAP rank of 80 was empirically chosen to produce best results. Table 5.4 summarizes the performance of the GMM-SVM based SV systems after NAP compensation. Marginal improvements in EER and MinDCF values are noticed, in comparison to the initial set of observations. The average EER reduction for the new set of SVMs in comparison to their earlier version (columns 2 and 4 of Table 5.4) are 1.07 % (Kl div) and 1.10 % (GUMI), respectively. The EER improvements in comparison to the baseline are 5.05 and 6.18 % for KL div and GUMI kernels, respectively. The improvements due to NAP are observed to diminish consistently with increasing SNR. This can be easily interpreted from the sequence of EER reductions for the KL div based systems given by 1.26, 1.13, 1.00 and 0.90 % for 0, 5, 7 and 10 dB SNRs, respectively. The same sequence for the GUMI based systems is 1.49, 1.18, 1.05 and 0.67 %.

Figure 5.5 shows the effect of NAP in the DET curves of the GMM-SVM based SV systems. The broken blue and red lines has been used to denote the NAP based GMM-SVM systems with KL div and GUMI kernels, respectively. No significant changes are apparent in the broken lines except for the consistent shift towards the origin which results in appropriate reduction in MinDCF values. In most cases, the KL div systems with NAP performs better than the GUMI based systems without NAP with an exception in the 10 dB SNR case. The overall improvement in average MinDCF values are 9.25×10^{-3} and 9.75×10^{-3} for SVM based systems with KL div and GUMI kernels, respectively.

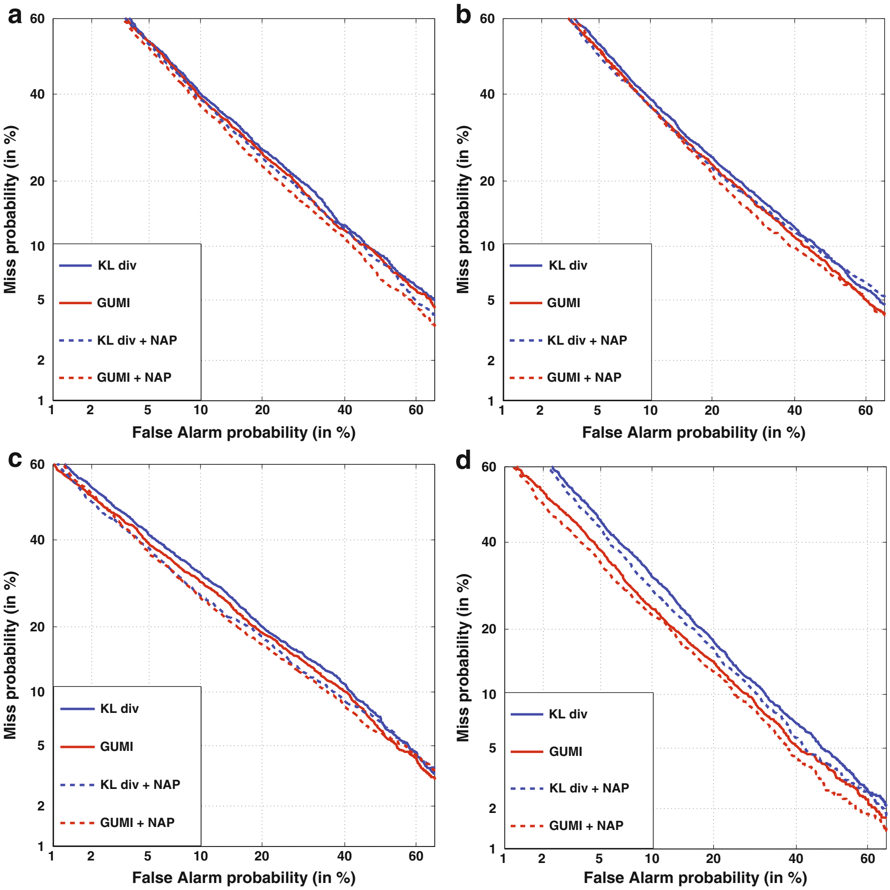

Fig. 5.5 DET plots showing the effect of NAP in the GMM-SVM based SV systems in varying background environment at (**a**) 0 dB (**b**) 5 dB (**c**) 7 dB and (**d**) 10 dB SNRs

Table 5.5 summarizes the relative EER values of the GMM-SVM based SV systems developed without and with NAP. An average relative EER of 20.19 and 24.89 % is obtained for the SVM based systems with KL div and GUMI kernels, respectively. However the improvement due to NAP is substantially limited with average relative EER increments of only 4.28 and 4.32 %, respectively for the aforementioned systems. The characteristics of (a) EERs and (b) Relative EERs of the GMM-SVM based systems at various SNRs in varying background environments has been shown in Fig. 5.6.

Table 5.5 Relative equal error rates for GMM-SVM based SV systems in varying background environments

SNR (dB)	Relative equal error rate (EER_R) (%) GMM-SVM KL div	GMM-SVM GUMI	GMM-SVM + NAP KL div	GMM-SVM + NAP GUMI
0	13.20	15.86	17.86	21.37
5	13.83	17.17	18.22	21.76
7	21.95	24.44	25.90	28.59
10	14.68	24.79	18.80	27.86

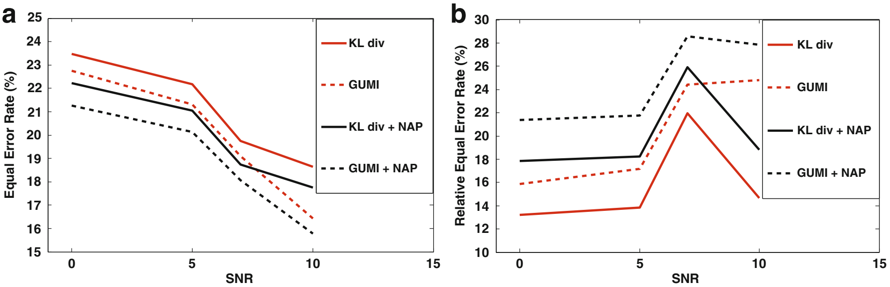

Fig. 5.6 (**a**) Equal error rates and (**b**) relative equal error rates of the GMM-SVM based SV systems at different SNRs in varying background environments

5.2 Utterance Partitioning for Improving GMM-SVM Based Speaker Verification Performance

The studies conducted in various types of noisy environments, as described in the previous section, unanimously indicates that the SV performance accuracy enhances with the use of GMM supervectors in conjunction with SVMs. However, it was also noticed that the performance improvements were not consistent across different noisy backgrounds at various SNR levels. In fact, fractional changes in EER values were observed in quite a number of cases e.g., uniform factory noise at 5 dB SNR. Besides, the use of GUMI kernels yielded marginal improvements in comparison to the standard KL div kernels, in most of the simulated environments. Contrary to expectations, the benefits of the complex NAP operations were also nominal. These phenomena suggested scope for further improvement in the standard SV system design. Instead of exploring alternative modeling methods, a number of inherent drawbacks in the existing method were addressed for a change. A few of such drawbacks can be highlighted as follows

- **Data imbalance:** A distinct aspect of the conventional SVM training method is that the number of background utterances (supervectors) vastly outnumber the number of enrolment utterances from a target speaker (typically one). This obviously leads to the generation of a larger number of support vectors in the majority class (background speakers) compared to the minority class (target speaker) causing a phenomenon called 'data imbalance' [11, 12]. As a consequence, the SVM decision boundary skews towards the minority class which causes high false rejection ('miss') rates during the kernel scoring in the evaluation phase (Eq. 5.8) of SV, unless the decision threshold is properly adjusted to compensate for the bias.

- **Mismatched utterance lengths:** The duration of training and test utterances plays a significant role in SV accuracy [13]. The amount of available training data in utterances determines the degree of MAP-adaptation of a GMM and thus affects the composition of the supervectors as discussed in Sect. 5.1. The difference in the enrolment and test utterance lengths (the former being considerably larger than the latter), can thus lead to statistical mismatches during the evaluation phase. In fact, prior studies have shown the benefits of matching training and test utterance durations for SV [14]. Additionally, recent studies have revealed that the discriminative power of fixed-size vectors used for representing variable length utterances saturates when the utterance length exceeds a threshold (typically 2 min) [15]. In such situations, the excess data can be utilized by generating new vectors rather than a single one.
- **Small sample-size problem:** In a typical training dataset, the number of speakers could be fairly large, but the number of available sessions per speaker are often quite limited. When the number of training speakers or the number of recording sessions per speaker are insufficient, numerical errors occur in estimating transformation matrices associated with the construction of supervectors (e.g., NAP), resulting in inferior performance (as noticed in Sect. 5.1.6). In machine learning literature, this is known as the 'small sample-size problem' [16, 17].

The various available strategies used to mitigate the effect of 'data imbalance' can be broadly categorized as (i) data processing approaches and (ii) algorithmic approaches. The family of methods in the first group tries to reduce the disproportionate ratio of support vectors in each class [18]. This can be done by (a) Over-sampling methods, where new training examples are generated from the existing minority class data [19, 20] (b) Under-sampling methods, where a subset of majority class examples are used to train individual SVMs [21, 22] and (c) a combination of Over-sampling and Under-sampling [12]. Under-sampling is usually not preferred for SV tasks since it causes loss of discriminative information whereas over-sampling methods are a trade-off between improved classification accuracy and increased computational load. The algorithmic approaches modify the classifier algorithm to counter data imbalance. Earlier methods assigned asymmetric misclassification costs to the positive and negative training examples [23] which was marginally effective since the Lagrange multipliers in both classes were scaled to satisfy a SVM constraint. Other methods modified the kernel function according to the data distribution which lead to complex training procedures [24].

The mismatch in utterance durations as highlighted earlier can be resolved by either using shorter length training/enrollment utterances or longer test utterances. In the former case, the major issue is to empirically determine an appropriate length of training utterances which can contribute towards MAP adaptation without sacrificing representative power. Lengthy test utterances as an alternative are usually not preferred for real-life applications. Handling the small sample-size problem is also subject to practical constraints such as availability of a co-operative set of speakers for multi-session recordings or requesting multiple enrolment utterances from client speakers etc.

As a solution to the aforementioned problems a synthetic data generation technique using partitioned utterances as proposed in [25], was applied in the present work. Specifically, the sequence of frames in an utterance were randomized followed by dividing it into a number of fixed-length sub-utterances which were individually used for supervector construction. The formal steps of the method, known as Utterance Partitioning with Acoustic Vector Resampling (UP-AVR), are briefly outlined as follows:

1. Given an enrollment utterance of a target speaker, the acoustic vectors (MFCCs) are computed and their sequence of occurrence (frame indices) in the utterance are randomized. This randomized sequence is then divided into N partitions (sub-utterances).
2. Steps 1 is repeated R times to produce RN sub-utterances.
3. Each of the sub-utterances produced in Step 2 together with the original utterance are individually used for supervector construction. Thus a total of $RN + 1$ target speaker supervectors are obtained.
4. Each background utterance is like-wise partitioned into N sub-utterances as given in Step 1. However, unlike the enrollment utterances, Step 2 is skipped and Step 3 is directly applied instead.
5. For B background utterances, a total of $B(N + 1)$ background supervectors are thus obtained.

Based on the length of available utterances in the present work, parameter values of $N = 2$ and $R = 3$ were empirically determined to produce best results [26]. For each target speaker, UP-AVR thus produced 7 target supervectors $(3 \times 2 + 1)$ and 1,065 background supervectors $(355 \times (2 + 1))$. The new set of labelled supervectors were subsequently used for training speaker-specific SVMs and evaluation, as discussed in Sects. 5.1.1 and 5.1.6, respectively.

Table 5.6 summarizes the effect of UP-AVR on the performances of the GMM-SVM based SV systems in uniform background environments. Drastic performance improvements are noticed compared to the initial set of results (refer Table 5.1). The average EER decrements across all three SNR levels, are 5.13, 6.64, 6.07 and 5.69 % for GMM-SVM (KL div) and 4.79, 6.57, 6.61 and 5.87 % for GMM-SVM (GUMI) in car, factory, pink and white noisy backgrounds, respectively. The magnitude of EER and MinDCF reductions are scaled considerably, thus resolving much of the inconsistencies noted earlier. In contrast to the fractional changes observed initially (see Table 5.1), performance improvements in factory noise backgrounds are observed to be the highest. The average EER improvements across all four types of noises are 7.35, 9.12 and 7.09 % for GMM-SVM (KL div) and 7.04, 9.44 and 7.35 % for GMM-SVM (GUMI) at 0, 5 and 10 dB SNRs, respectively. The GUMI kernels are observed to perform consistently better than the KL div kernels thereby asserting the significance of using covariance information for SV in degraded conditions. However, it is interesting to note that the performance

Table 5.6 Performance of the GMM-SVM based SV systems with UP-AVR in uniform background environments

		GMM-SVM				GMM-SVM with UP-AVR			
		KL div		GUMI		KL div		GUMI	
SNR	Noises	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0 dB	Car	14.32	0.063	13.82	0.058	11.25	0.043	11.02	0.042
	Factory	21.27	0.086	20.32	0.086	15.36	0.063	14.54	0.059
	Pink	21.23	0.087	19.92	0.085	14.86	0.061	14.36	0.059
	White	22.72	0.076	22.67	0.079	16.03	0.066	15.67	0.063
5 dB	Car	13.77	0.059	13.10	0.058	07.09	0.032	06.68	0.030
	Factory	20.87	0.085	19.92	0.084	12.24	0.048	11.74	0.047
	Pink	19.65	0.086	19.06	0.085	13.19	0.054	11.79	0.046
	White	20.96	0.074	20.73	0.072	15.36	0.065	14.27	0.061
10 dB	Car	12.24	0.047	11.38	0.046	06.59	0.031	06.23	0.029
	Factory	14.81	0.053	14.50	0.052	09.44	0.039	08.76	0.038
	Pink	15.67	0.058	15.72	0.057	10.21	0.043	08.72	0.037
	White	17.75	0.065	15.49	0.067	12.96	0.055	11.33	0.048
Clean		06.72	0.030	06.44	0.030	06.54	0.028	06.21	0.027

improvements due to UP-AVR in clean conditions are negligible which explains its effectiveness specifically for noisy backgrounds.

Figure 5.7 demonstrates the impact of UP-AVR in the DET plots of the GMM-SVM based SV systems in uniform noisy environments. A set of red and black lines has been used to denote the upgraded SV systems with KL div and GUMI kernels, respectively. The red and black curves can be easily distinguished from the set of blue and green curves which represents the initial set of GMM-SVM based systems. There is a wide margin of difference at all operating points of the new set of curves in comparison to the old ones. In most cases they are either entirely non-overlapping with the older ones or display the characteristic anti-clockwise rotation. A notable aspect of the UP-AVR based systems is that the performance upgradation at 0 dB SNR is comparable or even better than the initial systems at 10 dB SNR. A comparison of average MinDCF values across all 12 background environments in Table 5.6 show drastic improvements of 19×10^{-3} and 22.5×10^{-3} for GMM-SVM (KL div) and GMM-SVM (GUMI), respectively.

Table 5.7 summarizes the relative equal error rates of the GMM-SVM based SV systems developed using partitioned utterances in uniform noisy environment. The performance improvements due to UP-AVR are reflected by the dramatic increase in relative EERs. The average relative EERs in car, factory, pink and white noisy backgrounds are 51.94, 39.30, 44.76, 44.36 % for GMM-SVM (KL div) and 53.89, 42.65, 50.01, 48.59 % for GMM-SVM (GUMI), respectively. This is significantly higher than the initial set of relative EERs recorded in Table 5.2. The average improvements in relative EERs are 27.85 and 28.11 % for GMM-SVM (KL div) and GMM-SVM (GUMI), respectively.

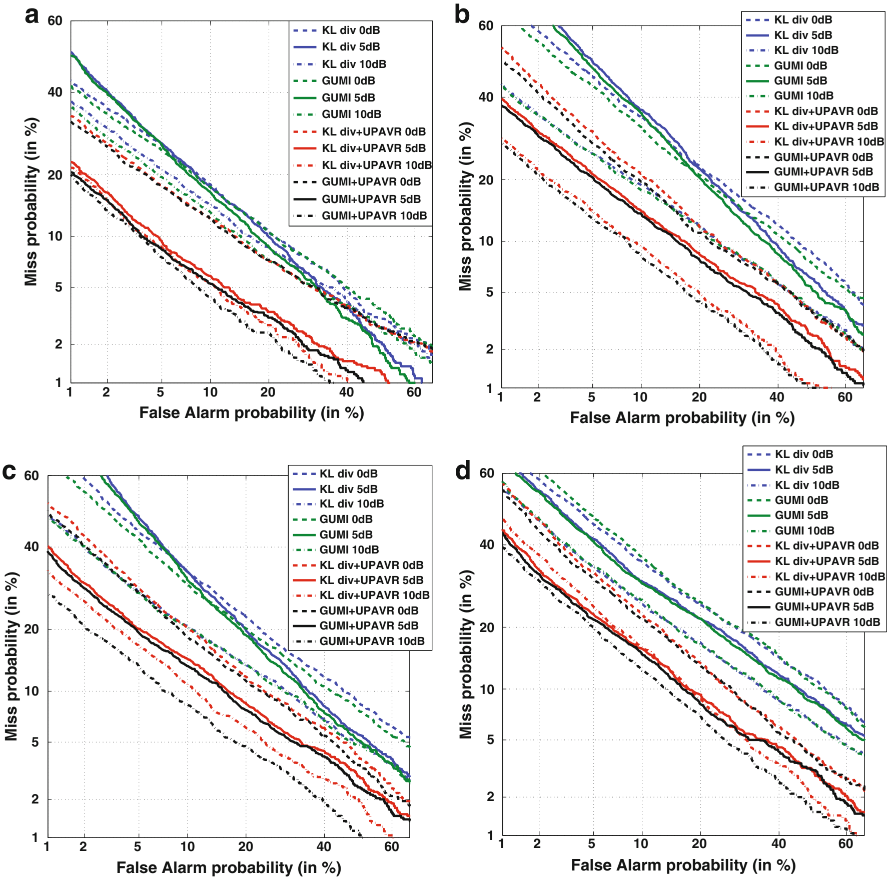

Fig. 5.7 DET plots showing the effect of UP-AVR on GMM-SVM based SV systems in uniform background environments with (**a**) car (**b**) factory (**c**) pink and (**d**) white noise

Figure 5.8 demonstrates the changes in (a) EERs and (b) Relative EERs of the GMM-SVM systems with UP-AVR, at various SNRs in uniform background environment. The characteristics of the EERs are in contrast to that observed earlier in Fig. 5.3. Specifically, the abrupt EER fluctuation at 5 dB SNR for factory and pink noises are much relaxed. However close resemblances (with Fig. 5.3) in the relative EER characteristics are noticed with an exception in case of factory noise. As usual an abrupt change in relative EER at 5 dB SNR is noticed with an exception in case of pink and white noise for GMM-SVM (KL div) where linearity in changes are retained.

The effect of UP-AVR was also studied for the SV systems developed in varying background environments. Just like the uniform background scenarios, significant

Table 5.7 Relative equal error rates for GMM-SVM based SV systems with UPAVR in uniform background environments

	Relative equal error rate EER_R (%)					
	SNR (0 dB)		SNR (5 dB)		SNR (10 dB)	
Noises	KL div	GUMI	KL div	GUMI	KL div	GUMI
Car	37.63	38.91	60.85	63.11	57.32	59.65
Factory	33.70	37.25	41.60	43.99	42.58	46.72
Pink	44.24	46.12	44.79	50.65	45.25	53.24
White	48.26	49.42	43.96	47.94	40.85	48.29

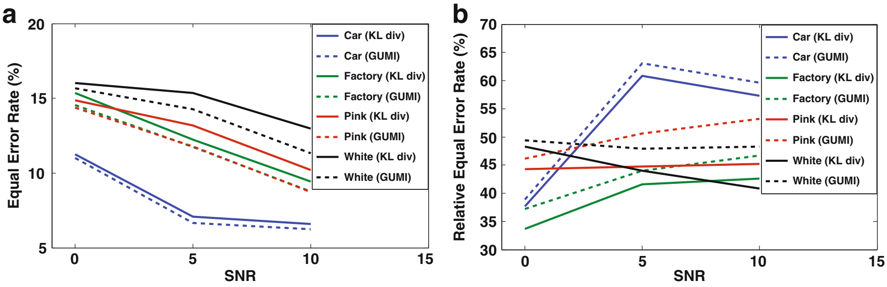

Fig. 5.8 (**a**) Equal error rates and (**b**) relative equal error rates of the GMM-SVM based SV systems with UP-AVR at different SNRs in uniform background environments

Table 5.8 Performance of the GMM-SVM based SV systems with UP-AVR in varying background environments

	GMM-SVM				GMM-SVM with UP-AVR			
SNR	KL div		GUMI		KL div		GUMI	
(dB)	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0	23.48	0.086	22.76	0.085	15.76	0.060	16.16	0.066
5	22.18	0.080	21.32	0.084	15.18	0.053	14.81	0.052
7	19.74	0.073	19.11	0.071	14.50	0.051	12.38	0.048
10	18.65	0.072	16.44	0.069	12.24	0.047	11.38	0.046

reduction in the error metrics are observed once again, in contrast to the initial set of system performances (without UP-AVR), as shown in Table 5.8. The EER reductions compared to the initial set of observations are 7.72, 7.00, 5.24, 6.41 % for GMM-SVM (KL div) and 6.60, 6.51, 6.73, 5.06 % for GMM-SVM (GUMI) at 0, 5, 7 and 10 dB SNRs, respectively. The two SVM kernels show different behavior in terms of EER changes with a slight anomaly noticed at 0 dB SNR where the KL div kernel performs better than the GUMI kernel. The effect of UP-AVR also appears to be more prominent in case of KL div kernels which shows an average EER improvement of 6.59 % across all SNR levels in comparison to 6.23 % for the GUMI kernel.

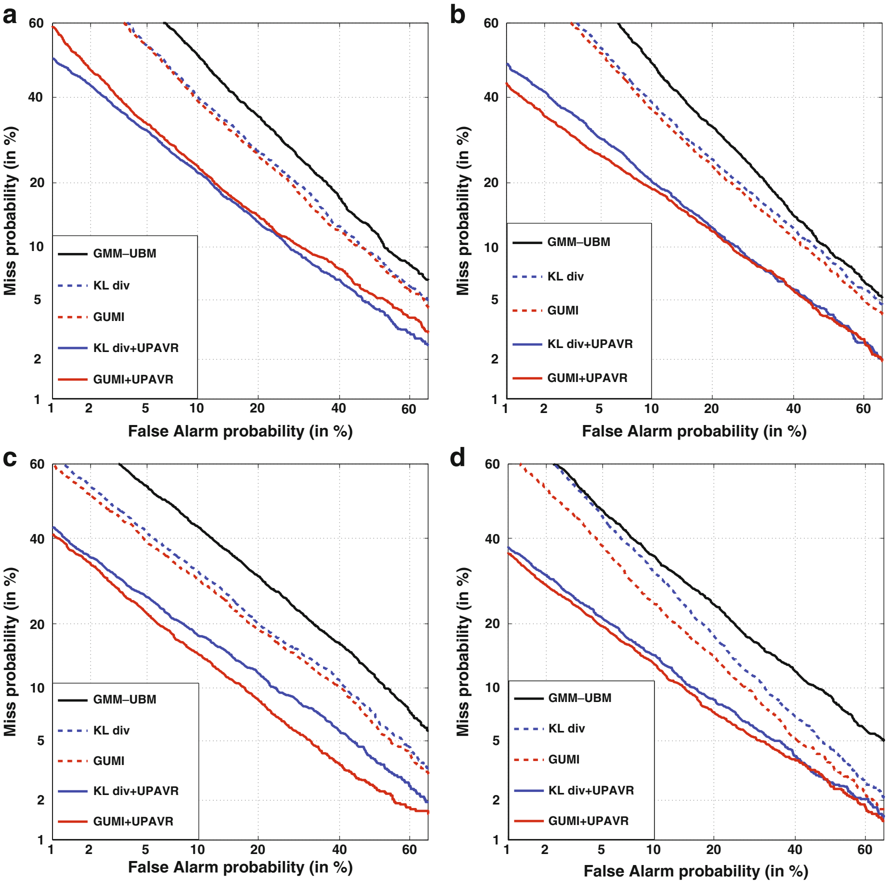

Fig. 5.9 DET plots showing the effect of UP-AVR on GMM-SVM based SV systems in varying background environments at (**a**) 0 dB (**b**) 5 dB (**c**) 7 dB and (**d**) 10 dB SNRs

Figure 5.9 demonstrates the impact of UP-AVR in the DET plots of the GMM-SVM based SV systems in varying background environments. The set of solid blue and red lines denote the UP-AVR based GMM-SVM systems with KL div and GUMI kernels, respectively. The broken lines of same colors represent the initial systems developed in the same backgrounds while the black line represents the baseline. As usual a wide margin is noticed between the solid and broken set of curves. Unlike the initial set of GMM-SVM systems (see Fig. 5.4), dissimilarities are observed in the curves corresponding to the two SVM kernels. In most cases, the red and blue curves show distinct behavior. Apart from the overall shift towards the origin, anti-clockwise rotations in the red and blue curves are prominently noticed in

Table 5.9 Performance of the GMM-SVM based SV systems with UP-AVR and NAP compensation in varying background environments

SNR	GMM-SVM + UP-AVR				GMM-SVM + UP-AVR + NAP			
	KL div		GUMI		KL div		GUMI	
(dB)	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
0	15.76	0.060	16.16	0.066	13.37	0.058	13.62	0.063
5	15.18	0.053	14.81	0.052	13.23	0.051	12.47	0.048
7	14.50	0.051	12.38	0.049	13.10	0.049	11.29	0.046
10	12.24	0.047	11.38	0.046	11.21	0.044	10.32	0.043

case of 5 and 10 dB SNRs. The resultant improvement in MinDCF values averaged across all SNRs are 25.00×10^{-3} and 24.25×10^{-3} for KL div and GUMI kernels, respectively.

As mentioned earlier in Sect. 5.2, the UP-AVR strategy was adopted to alleviate a set of three highlighted drawbacks of the conventional GMM-SVM based systems. However, it is difficult to conclude the degree of impact UP-AVR exercises on each of them. In most cases one may rely on the joint improvement of all three problems, without specifically knowing each of them. In order to demonstrate the specific utility of UP-AVR towards mitigating the small sample-size problem, the partitioned enrollment utterances (supervectors) were subjected to NAP transformation prior to SVM training.

Unlike its earlier version, the supervector matrix V constructed in the Step 3 of the NAP algorithm, now had an expanded size of 2,800 × 39,936 due to the impact of UP-AVR on the target speaker utterances. All the training supervectors were subjected to NAP transformation prior to SVM training with required strategies for maintaining feasibility in large matrix operations as discussed in Sect. 5.1.6.1.

Table 5.9 summarizes the performance of the GMM-SVM based SV developed using UP-AVR followed by NAP compensation in varying background environments. In contrast to the initial set of observations (see Table 5.4), a larger average EER reduction compared to the baseline (i.e., 12.26 % (KL div) and 13.06 % (GUMI)) is noticed across all four SNR levels. The additional improvements due to NAP over UP-AVR are 2.39, 1.95, 1.40, 1.03 % and 2.54, 2.34, 1.09, 1.06 % at 0, 5, 7 and 10 dB SNRs for GMM-SVM (KL div) and GMM-SVM (GUMI), respectively. The effect of NAP compensation is observed to be more prominent in case of the GUMI kernels.

Figure 5.10 demonstrates the effect of NAP on the DET plots of the GMM-SVM based systems developed in varying background environments. The color coding for representing each system is the same as that in Fig. 5.9. The broken lines of each color have been used to denote the corresponding systems with NAP transformation. The set of blue and green lines seen earlier in Fig. 5.5, are included again for studying the overall comparison of the various systems. The behaviour of the NAP-based curves are quite similar in shape and alignment to the initial systems except for a larger margin of difference from them at all operating points. Expectedly, a significant improvement in MinDCF values are observed

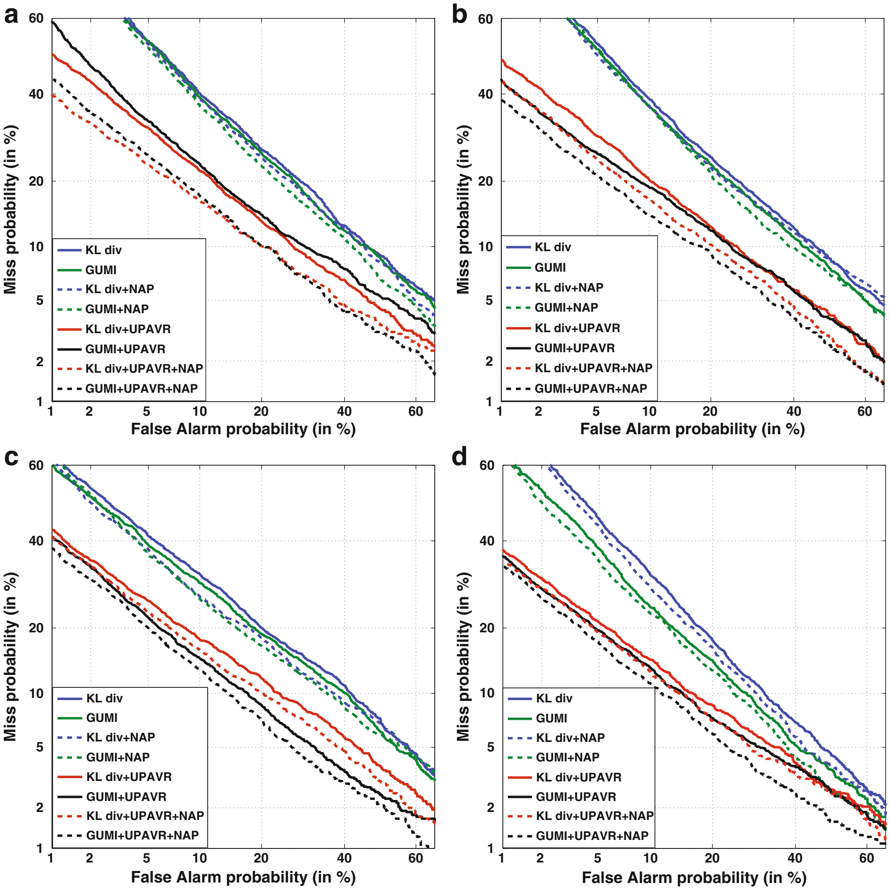

Fig. 5.10 DET plots showing the effect of UP-AVR and NAP on GMM-SVM based SV systems in varying background environment at (**a**) 0 dB (**b**) 5 dB (**c**) 7 dB and (**d**) 10 dB SNRs

in comparison to the earlier NAP-based SV systems with average reduction of 26.00×10^{-3} and 25.75×10^{-3} across all SNRs, for KL div and GUMI kernels, respectively.

Table 5.10 summarizes the relative EERs of the various GMM-SVM based SV systems developed in varying background environments. The average relative EERs across all SNRs, are 42.36 and 45.43 % for UP-AVR based GMM-SVM systems with KL div and GUMI kernels, respectively. The corresponding values with an additional NAP application are 49.02 and 52.34 %. The benefits of utterance partitioning can be observed from the significant average improvements of 26.68 and 25.45 % relative EER rates for the two types of NAP based GMM-SVM based systems.

Table 5.10 Comparison of relative equal error rates for GMM-SVM based SV systems in varying background environments

SNR	Relative equal error rate EER_R (%)							
	GMM-SVM		NAP		UP-AVR		UP-AVR + NAP	
(dB)	KL div	GUMI	KL div	GUMI	KL div	GUMI	KL div	GUMI
0	13.20	15.86	17.86	21.37	41.74	40.26	50.57	49.65
5	13.83	17.17	18.22	21.76	41.03	42.46	48.60	51.55
7	21.95	24.44	25.90	28.59	42.67	51.05	48.20	55.36
10	14.68	24.79	18.80	27.86	44.01	47.94	48.72	52.79

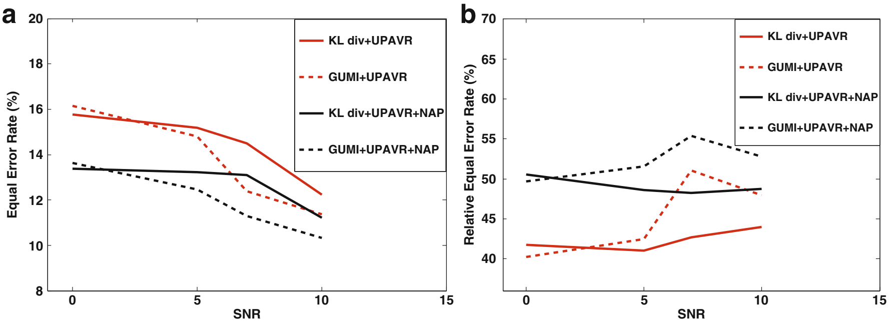

Fig. 5.11 (**a**) Equal error rates and (**b**) relative equal error rates of the GMM-SVM based SV systems with UP-AVR at different SNRs in varying background environments

Figure 5.11 demonstrates the changes in (a) EERs and (b) Relative EERs of the GMM-SVM systems with UPAVR, at various SNRs in varying background environments. Notable changes in the behavior of the red and black curves are observed in contrast to those in Fig. 5.6. The abrupt EER and relative EER fluctuations at 7 dB SNR, initially observed in Fig. 5.6 are now relaxed for the GMM-SVM (KL div) system where NAP application makes an anomalous change. The GUMI based GMM-SVM systems however show similar behavior with and without NAP applications which is characterized by consistent increase in relative EER values with increasing SNRs with an abrupt decrement at the 10 dB SNR level.

5.3 Total Variability Modeling for Speaker Verification

The significance of the GMM-SVM methods for SV in noisy environment was explored through an extensive set of empirical studies discussed in Sects. 5.1 and 5.2, respectively. Despite the drastic performance enhancements achieved using the UP-AVR strategy, few typical limitations of the developed SV systems can be highlighted. Firstly, the large size of the GMM supervectors are a practical constraint in terms of their memory consumption and computational costs (e.g., SVM training, NAP transformation). Secondly, despite UP-AVR the performance improvements of the SV systems developed in extremely degraded conditions in the uniform background environments were comparatively lower. Specifically, the average relative EERs of the SV systems across the four different backgrounds were 41.95 and 49.61 % at 0 and 5 dB SNRs in contrast to a larger value of 54.23 % at 10 dB SNR. Even the average EERs at 0 and 5 dB i.e., 14.13 and 11.54 % were significantly larger than those at 10 dB (9.28 %). Individual EERs were observed to be typically high for factory, pink and white noisy backgrounds. These factors suggested the use of alternative robust speaker modeling methods for further improvement in performance accuracy. Specifically a state-of-the-art low dimensional representation of GMM supervectors, commonly known as identity vectors or 'i-vectors' [7], was used for developing SV systems. In the remaining part of this section, the details of i-vector extraction, its application and evaluation in the present work are discussed.

5.3.1 i-Vector Extraction

Total variability modeling [7] is based on projecting large dimensional supervectors in a low dimensional subspace (known as 'total variability' space) which supposedly contains both channel and session information. Specifically, a GMM mean supervector M is represented as

$$M = m + Tw \tag{5.11}$$

where m is a speaker/channel independent supervector (i.e., the UBM mean supervector), T is low-rank rectangular matrix whose columns consists of eigenvectors of the total variability covariance matrix with largest eigenvalues. w is a random vector having standard Normal distribution, called i-vector. The total variability matrix (T) is learned offline, using probabilistic principal component analysis (PPCA) [4] on a development dataset [27, 28]. Estimation of i-vectors from a set of utterances requires initial computation of a set of Baum-Welch statistics followed by a set of matrix operations involving them. Given a sequence of D-dimensional acoustic vectors $\{x_1, x_2, \ldots x_{\mathbf{T}}\}$ of an utterance X with $\mathbf{T}$ frames, the Baum-Welch statistics are calculated as

$$N_i = \sum_{t=1}^{\mathbf{T}} p(i|x_t, \lambda)$$

$$F_i = \sum_{t=1}^{\mathbf{T}} p(i|x_t, \lambda)(x_t - m_i)$$

where $p(i|x_t, \lambda)$ is the posterior probability of the ith Gaussian component of a UBM λ having total M components, which generates vector x_t. The mean of the same component is given by m_i. N_i and F_i are known as the zeroth order and mean-shifted first order sufficient statistics, respectively.

Given the trained T matrix and the set of Baum-Welch statistics, the i-vector extracted from utterance X is calculated as

$$w = (I + T^T \Sigma^{-1} N(X) T)^{-1} . T^T \Sigma^{-1} F(X) \tag{5.12}$$

where Σ and $N(X)$ are block diagonal matrices of size $(MD \times MD)$ whose diagonal blocks consist of the UBM covariance matrices Σ_i $(i = 1, 2, \ldots M)$ and identity matrices weighted with the zeroth order statistics $N_i I_{D \times D} (i = 1, 2, \ldots M)$, respectively. F(X) is a supervector obtained by stacking the mean-shifted first order statistics F_i $(i = 1, 2, \ldots M)$ and I is an identity matrix of size $(MD \times MD)$. Total variability modeling is generative in nature, however they can be integrated with a discriminative framework using SVMs [29, 30]. The detailed procedure of training the T matrix has been outlined in Appendix D.

5.3.2 SVM Training

Since i-vectors are fixed-length vectors representing variable length utterances, they can be used to train SVMs using sequence kernels as discussed in Sect. 5.1.1. It was investigated in [29], that the best result in i-vector frameworks are produced by using a cosine kernel function for training the SVMs, which can be defined for two input i-vectors w_1 and w_2 as

$$k(w_1, w_2) = \frac{\langle w_1, w_2 \rangle}{\|w_1\| \|w_2\|} \tag{5.13}$$

where $\langle ., . \rangle$ and $\|.\|$ denote the inner product and L2-norm, respectively. The cosine kernel normalizes the linear kernel by the norm of both i-vectors. It considers only the angle between the two i-vectors and not their magnitudes. It is believed that non-speaker information (such as session and channel) affects the i-vector magnitudes, removing which improves the robustness of the i-vector system.

5.3.3 Inter-session Compensation

Since the total variability subspace contains both speaker and session variability information, i-vectors extracted from it are usually subjected to session compensation prior to SVM training. Two common session compensation techniques used in the i-vector framework are discussed as follows

5.3.3.1 Linear Discriminant Analysis (LDA)

LDA [4] projects the i-vectors to a set of orthogonal axes for minimizing within-class variance and maximizing between-class variance. In the i-vector framework, all i-vectors extracted from a speaker constitute a particular class. The projection matrix A is composed of eigenvectors v having the highest eigenvalues λ, obtained by solving the following generalized eigen decomposition problem

$$B_S v = \lambda W_S v \tag{5.14}$$

where B_S, W_S are the between-class and within-class covariance matrices given by

$$B_S = \sum_{s=1}^{S} (\mu_s - \mu)(\mu_s - \mu)^T \tag{5.15}$$

$$W_S = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (w_i - \mu_s)(w_i - \mu_s)^T \tag{5.16}$$

where S is the total number of speakers, n_s is the total number of utterances from the sth speaker, μ_s is the mean of all i-vectors (w_i) from speaker s given by $\mu_s = \frac{1}{n_s} \sum_{i=1}^{n_s} w_i$ and μ is the global mean of all i-vectors generally considered to be a null vector due to their standard normal distribution. The number of columns of the matrix A (i.e., LDA order) are determined empirically to produce best results. The LDA-modified cosine kernel function for two input i-vectors w_1 and w_2 is given by

$$k(w_1, w_2) = \frac{(A^T w_1)^T (A^T w_2)}{|A^T w_1||A^T w_2|} \tag{5.17}$$

5.3.3.2 Within-Class Covariance Normalization (WCCN)

WCCN, proposed in [31] aims to set upper bounds on the error metrics (‘miss’ and ‘false alarm’) by normalizing the SVM kernels. Application of WCCN in the i-vector framework requires projecting the i-vectors to a space specified by the square-root of the inverse of the within-class covariance matrix. Specifically, the projection matrix B is obtained by Cholesky decomposition of the inverse of the within-class covariance matrix (Eq. 5.16) as follows

$$W_S^{-1} = BB^T \tag{5.18}$$

The WCCN-modified cosine kernel function for two input i-vectors w_1 and w_2 is given by

$$k(w_1, w_2) = \frac{(B^T w_1)^T (B^T w_2)}{|B^T w_1||B^T w_2|} \tag{5.19}$$

5.3.4 Score Calculation

Two types of i-vector evaluation methods, namely Cosine Distance scoring and SVM Kernel scoring, has been proposed in past [29]. The former one is applied in the default generative modeling framework while the latter for the discriminative SVM framework.

5.3.4.1 Cosine Distance Scoring (CDS)

CDS is a fast scoring method commonly applied in i-vector frameworks. As the name suggests, it is simply the cosine distance between a pair of i-vectors representing a claimant's test utterance (w^{test}) and the claimed target speaker utterance (w^{target}), respectively as given by

$$S_{cos} = \frac{< w^{test}, w^{target} >}{|w^{test}||w^{target}|} \tag{5.20}$$

5.3.4.2 SVM Kernel Scoring

The SVM scoring is exactly similar to the one already discussed in Sect. 5.1.4. The advantage of SVM scoring is that the contribution of individual speakers towards the verification scores can be optimally weighted by the Lagrange multipliers of the target speakers SVM. Given a trained target speaker SVM and the test i-vector w^{test}, the score is calculated as

$$S_{SVM} = \sum_{t=1}^{T} \alpha_t K(w^t, w^{test}) - \sum_{i=1}^{B} \alpha_i K(w^i, w^{test}) + d \tag{5.21}$$

where w^t and w^i are the sequence of support vectors corresponding to the target and background speaker classes as learned during SVM training. α_t and α_i are the non-zero Lagrange multipliers of the corresponding classes. T and B are the total number of support vectors in each class, d is a bias term and K is the cosine kernel (Eq. 5.13).

5.3.5 Experimental Setup

Figure 5.12 shows a block diagram of the i-vector based SV system. The SV systems were developed using the set of noise-degraded training and test utterances of NIST-SRE-2003 in uniform background environment (see Sect. 5.1.5.1) at 0 and 5 dB SNRs, as discussed in Sects. 5.3.1 and 5.3.3. A development data comprising

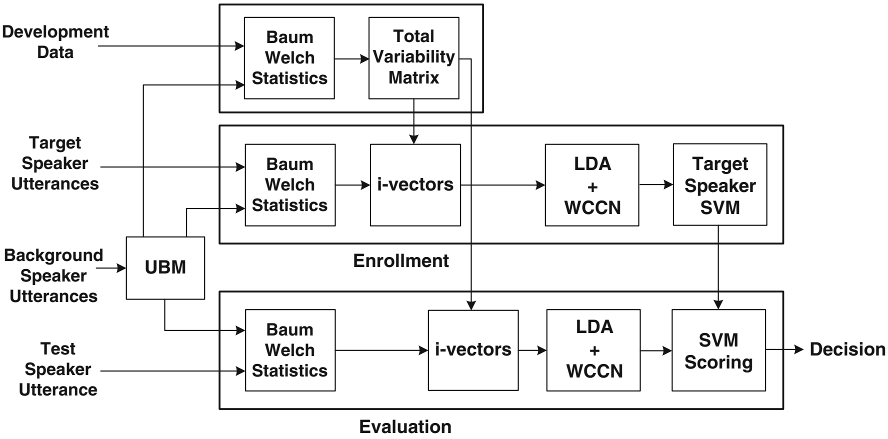

Fig. 5.12 Block diagram of the combined SVM and total variability modeling framework for speaker verification

1,572 utterances from the SwitchBoard phase II corpora and 400 utterances from NIST-SRE-2004 database was used for training the total variability and channel compensation matrices (see Appendix D). The T-matrix rank of 400 was chosen empirically and i-vectors were extracted from all utterances as discussed in Sect. 5.3.1. The low dimension of i-vectors facilitated convenient application of LDA and WCCN, the projection matrices for which were designed as discussed in Sect. 5.3.3. A LDA order of 300, was empirically determined to produce best results. All the i-vectors were subjected to session compensation prior to model building. A discriminative framework (combined i-vector and SVM) for classification was used instead of the conventional generative i-vector modeling in favor of utilizing the benefits of UP-AVR and SVM scoring as shown in [15]. The labelled i-vectors extracted from enrollment and background speaker's utterances were subjected to speaker specific SVM training. During the evaluation phase, the noisy test utterances (i-vectors) were evaluated against the target speaker models (SVMs) according to NIST-2003 primary task, using the SVM scoring method as discussed in Sects. 5.1.6 and 5.3.4.2, respectively. The experiments were repeated using partitioned utterances with UP-AVR parameters $N = 2$ and $R = 3$.

Table 5.11 summarizes the performance of the i-vector based SV systems developed in uniform noisy environments at 0 and 5 dB SNR. While the error metrics show considerable performance improvements compared to the GMM-SVM based systems in individual noisy backgrounds, it is interesting to note that the GMM-SVM based systems developed using UP-AVR performs better than the i-vector models developed without UP-AVR. This can be deduced from a comparison of the GMM-SVM based SV systems in (Table 5.6). The average EER reductions across both SNRs compared to the default GMM-SVM based SV systems are 3.52, 6.12, 4.90 and 5.51% for car, factory, pink and white noisy backgrounds,

Table 5.11 Performance of the i-vector based SV systems in uniform background environments at 0 and 5 dB SNR

	SNR (0 dB)				SNR (5 dB)			
	Without UP-AVR		With UP-AVR		Without UP-AVR		With UP-AVR	
Noises	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
Car	12.10	0.051	10.57	0.047	08.31	0.039	06.05	0.028
Factory	16.17	0.068	12.33	0.055	12.78	0.057	09.03	0.042
Pink	16.72	0.071	12.83	0.054	13.42	0.058	10.16	0.044
White	17.39	0.073	14.27	0.059	15.13	0.063	11.88	0.051

respectively. However, GMM-SVM with UP-AVR performs slightly better than the current systems with average reduced EERs of 1.23, 1.01, 1.52 and 0.93% at the corresponding environments across both SNRs. This phenomenon once again establishes the significance of UP-AVR in enhancing SV performances in noisy conditions. The superiority in i-vector performance accuracies are restored by incorporating UP-AVR in its framework. Comparison amongst the UP-AVR based systems (see Table 5.6) reveals average EER reductions of 0.70, 2.79, 2.06, 2.26% in car, factory, pink and white noisy backgrounds, respectively.

Figure 5.13 shows the DET plots of the i-vector based SV systems developed in uniform noisy environments. As usual a shift towards the origin is observed in the curves corresponding to the UP-AVR based systems (represented by broken lines) suggesting consistent reduction in MinDCF and EER values across each noisy background. The effect of UP-AVR at 0 dB SNR is apparently more prominent in case of the colored noises. Unlike the GMM-SVM based systems (see Fig. 5.7), no significant change in slope or rotation of the curves are noticed. The average improvements in MinDCF values of the i-vector based SV systems (with and without UP-AVR) in comparison to the corresponding GMM-SVM based SV systems (see Tables 5.1 and 5.6) are 2.94×10^{-3} and 1.82×10^{-3}, respectively.

Despite the apparent performance improvements achieved by the i-vector based SV systems, a typical aspect to be noticed is that UP-AVR results in a moderate decrement of only 3.10 % average EER. Similar observations were earlier recorded for the GMM-SVM based systems (see Table 5.6) which had shown 5.39 % EER reductions at 0 dB SNR in contrast to larger improvements at 5 and 10 dB SNRs. This phenomenon indicates the obvious increase in classification errors due to high noise strength. A typical drawback of the standard UP-AVR algorithm can also be highlighted in this context. Specifically, all speaker's utterances are partitioned irrespective of the role they play towards classification. This could be detrimental towards SV performance e.g., partitioning a speaker's utterance which was originally misclassified could lead to additional misclassifications apart from increased computational load. In order to alleviate these two problems in parallel, a novel boosting algorithm is proposed to train multiple SVM classifiers on the noisy dataset, the utterances in which are selectively used for partitioning. Subsequent sections provide the details of the boosting algorithm followed by their implementation in the i-vector based SV framework.

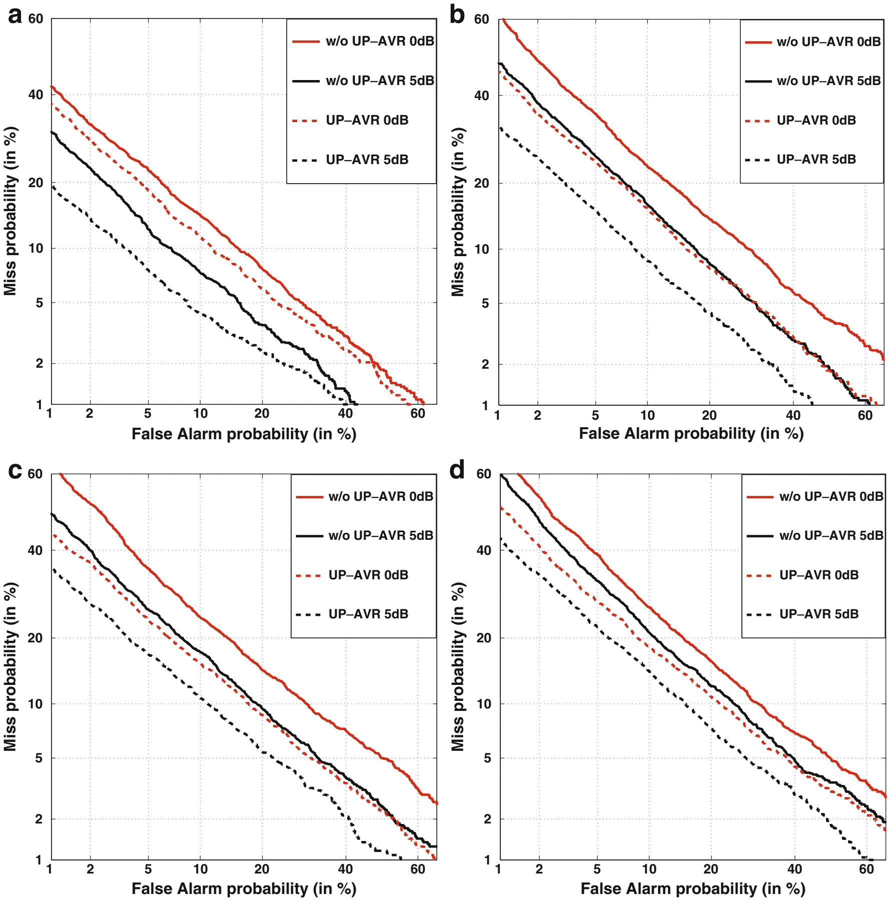

Fig. 5.13 DET plots of the *i-vector* based SV systems in uniform background environments with (**a**) car (**b**) factory (**c**) pink and (**d**) white noise at 0 and 5 dB SNR

5.4 Adaptive Boosting for Improved Speaker Verification Performance in Noisy Environments

Adaptive Boosting (AdaBoost) iteratively enhances the predictive accuracy of a sequence of weak classifiers (ensemble), each of which is trained on a dataset adaptively sampled according to the training error of the classifier in the previous iteration [32]. The final decision is based on a weighted voting of the individual classifiers in the ensemble. In recent past, boosting has been applied effectively for robust SV tasks [33]. Prior art also demonstrates the benefits of combining ensemble learning with data balancing [20, 34]. A novel combination scheme of the AdaBoost algorithm with a synthetic data generation technique using UP-AVR

[25], is proposed in the present work. The approach is motivated by the Databoost-IM algorithm proposed in [35]. The aim is to improve the predictive accuracy of both minority (target speaker) and majority (background speakers) classes while emphasizing on the misclassified examples in the minority class.

5.4.1 Proposed Boosting Algorithm (DataBoost-UP)

Conventional boosting algorithms emphasize on the misclassified (hard) training instances at each iteration by adaptively increasing their sampling weights. Classifiers trained in successive iterations concentrate on these instances with high weights. Since all misclassified examples are equally weighted, it doesn't compensate for the bias towards the majority class in imbalanced datasets. The aim of integrating data generation with the boosting algorithm is to alleviate the learning algorithm's bias towards the majority class while retaining focus on the hard training instances. Unlike the DataBoost-IM algorithm [35], in the proposed algorithm (DataBoost-UP) the data (i-vectors) is synthesized using the utterance partitioning technique [25] instead of random generation of attribute values in the [min,max] interval. Both the minority (target speaker) and majority (background speakers) classes are oversampled to prevent overemphasis on the hard instances of the minority class. The proposed algorithm is used to create an ensemble of SVM classifiers.

Algorithm DataBoost-UP

```
Input:
        Training data set $\{(x_i, y_i)\}_{i=1}^{N}, \quad y_i \in \{-1, +1\}$

        Weak SVM classifiers $h_t$ where $t = \{1, 2 \ldots, T\}$

Initialize: Sampling weight distribution $D_1(i) = 1/N \;\; \forall i = \{1, 2, .., N\}$

Do for t ← 1 to T

1. Identify the hard examples in the training set.
2. Generate new data from these examples by UP-AVR. Add them to the original training set.
3. Adjust the sampling weight distribution of both classes in the new training set.
4. Learn weak SVM $h_t$ on the new training set sampled according to the modified distribution.
5. $\epsilon_t \leftarrow \sum_{i=1}^{N} D_t(i) I(h_t(x_i) \neq y_i)$. If $\epsilon_t > 0.5$ set T = t-1 and abort loop.
6. $\alpha_t \leftarrow \frac{1}{2} \log\{(1 - \epsilon_t)/(\epsilon_t)\}$
7. $D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} \exp(-\alpha_t h_t(x_i) y_i)$ where $Z_t = \sum_{i=1}^{N} D_t(i) \exp(-\alpha_t h_t(x_i) y_i)$

Output: SVM ensemble $h_{final} = \sum_{t=1}^{T} \alpha_t h_t$
```

The predictive accuracy of the ensemble is guaranteed to improve in each iteration provided the training error of the weak SVM classifier in the previous iteration is less than 0.5 (upper bound). The ensemble training error decreases in successive iterations. At the end of a pre-determined number of iterations, the algorithm converges with no further decrement in the ensemble training error. Steps 1, 2 and 3 of the proposed algorithm are elaborated in the next three sections.

5.4.1.1 Identifying Hard Training Examples

The hard training examples are identified as follows.

1. All the instances in the training set are arranged in descending order of their sampling weights.
2. The top N_{train} number of instances of the training set are selected as hard examples where:
   $N_{train} = \epsilon_t \times N$,
   ϵ_t = weighted training error of a SVM in the tth iteration of boosting
   N = total number of instances in the original training set.
3. Let $N_{train} = N_{maj} + N_{min}$ where:
   N_{maj} = number of instances from majority class, N_{min} = number of instances from minority class.
   These training utterances are subjected to utterance partitioning as discussed in Sect. 5.4.1.2

5.4.1.2 Synthesizing Data Using Utterance Partitioning

The UP-AVR algorithm (discussed in Sect. 5.2) is applied for data generation, as follows

1. Given each of the N_{min} target speaker utterance, its acoustic vectors are computed and their sequence of occurrences in the utterance are randomized. This randomized sequence is then divided into P partitions (sub-utterances).
2. Step 1 is repeated R times. Together with the original full-length utterance, a total of $RP + 1$ utterances generated from each enrollment utterance are individually subjected to i-vector construction.
3. Similarly, each background speaker's utterances are divided into P partitions. For N_{maj} background speakers we thus have $N_{maj}(P + 1)$ utterances. Background i-vectors are constructed from each of these utterances.

5.4.1.3 Balancing Weights of Majority and Minority Classes

The aim of weight balancing is to minimize the difference between the total sampling weight of each class in an imbalanced dataset. This forces the boosting algorithm to focus on both the hard as well as rare training examples. The sampling

Table 5.12 Comparison of the effects of UP-AVR and Databoost-UP on the performances of i-vector based SV systems in uniform background environments at 0 and 5 dB SNRs

	SNR (0 dB)				SNR (5 dB)			
	UP-AVR		DataBoost-UP		UP-AVR		DataBoost-UP	
Noises	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF	EER (%)	MinDCF
Car	10.57	0.047	08.22	0.037	06.05	0.028	04.83	0.021
Factory	12.33	0.055	10.93	0.048	09.03	0.042	07.14	0.032
Pink	12.83	0.054	11.21	0.047	10.16	0.044	08.13	0.035
White	14.27	0.059	13.05	0.053	11.88	0.051	10.03	0.043

weight of each hard instance is divided by the number of new instances generated from it. All generated instances are uniformly assigned the divided weight. At the end the weights are rebalanced across the entire set of newly generated instances. If the total weight of the majority class (W_{maj}) exceeds that of the minority class (W_{min}) then each minority weight is scaled by a factor W_{maj}/W_{min}. For the vice-versa condition, each majority weight is scaled by a factor W_{min}/W_{maj}.

5.4.2 *Performance Evaluation*

The data used for experimental setup is identical to that described in Sect. 5.3.5. The i-vectors extracted from the partitioned target speaker utterances from each noisy dataset were used for training a SVM ensemble using the DataBoost-UP algorithm. Additionally, new data was generated in each iteration of the boosting algorithm with partitioning parameters values of $P = 2$ and $R = 1$ as discussed in Sect. 5.4.1.2. The number of boosting iterations ranging from 5 to 10 was empirically determined to appropriately lower the ensemble training error. During the evaluation phase, each test utterances were scored against 11 target speaker SVM ensemble. Given a noisy test utterance (i-vector) w^{test}, the Kernel scoring was obtained as a weighted linear combination of the scores obtained from individual classifiers of the target speaker ensemble as follows:

$$Score(w^{test}) = \sum_{i=1}^{T} \alpha_i (\sum_{j=1}^{L} \beta_{i,j} t_{i,j} K(w^{i,j}, w^{test}) + d_i)$$

where T is the size of the ensemble. α_i is the weight of the ith SVM classifier in the ensemble as calculated in Step 6 of the DataBoost-UP algorithm. $w^{i,j}$, $\beta_{i,j}$ and $t_{i,j} \in \{-1, +1\}$ are the sequence of L learned support vectors, the non-zero Lagrange multipliers and the actual class labels, respectively for the ith SVM classifier in the ensemble, d_i is the bias term and K is the cosine kernel function.

Table 5.12 summarizes the comparative performances of DataBoost-UP and UP-AVR method in the i-vector framework for the SV systems developed in uniform

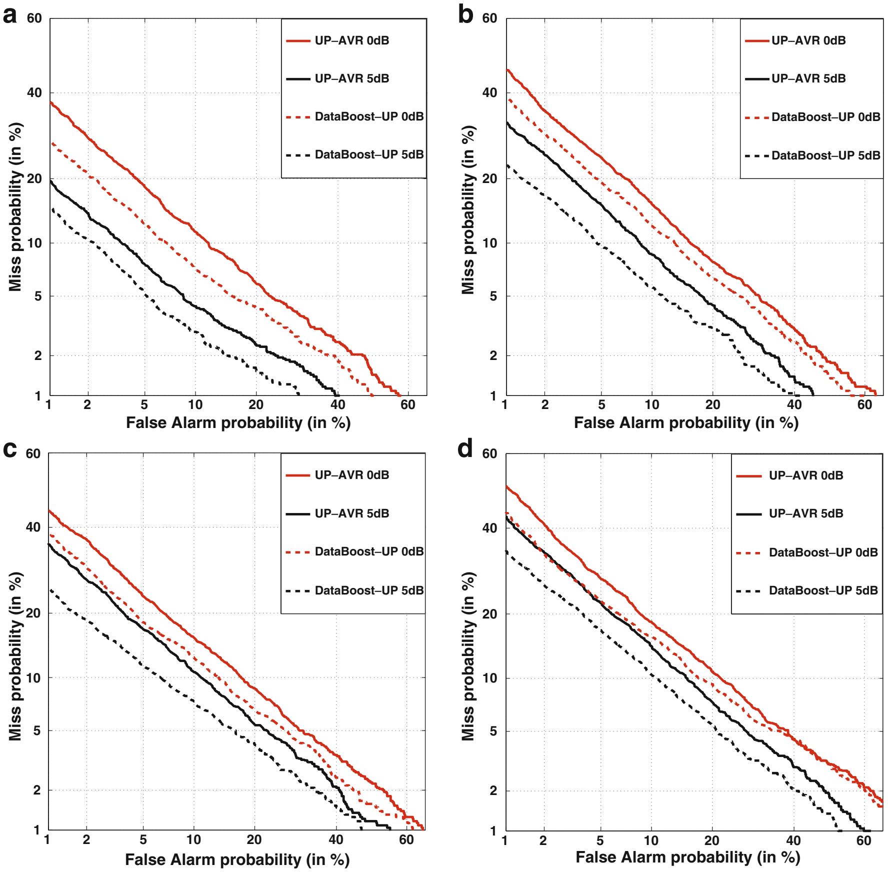

Fig. 5.14 DET plots showing the effect of UP-AVR and DataBoost-UP on the *i-vector* based SV systems in uniform background environment with (**a**) car (**b**) factory (**c**) pink and (**d**) white noise at 0 and 5 dB SNR

background environments. A consistent performance improvement is noticed in the boosted i-vector framework across all noisy backgrounds, in comparison to the UP-AVR based system at both SNR levels. The individual EER reductions are 2.35, 1.40, 1.62 and 1.22 % at 0 dB SNR and 1.22, 1.89, 2.03, 1.85 % at 5 dB SNR for car, factory, pink and white noisy backgrounds, respectively. Thus an additional average EER reduction of 1.70 % across all environments is achieved on top of the initial improvement (see Table 5.11) of 3.12 % due to UP-AVR.

Figure 5.14 shows the DET plots of the i-vectors based SV systems using (a) UP-AVR and (b) DataBoost-UP, respectively. Interestingly, the nature of improvement in each curve is similar to those observed in the Fig. 5.13. There are no apparent rotation in the curves apart from the overall shift towards the origin characterized by the corresponding changes in detection costs. In contrast

Table 5.13 Comparison of relative equal error rates of SV systems developed in uniform background environments at 0 and 5 dB SNRs

		Relative equal error rate EER_R (%)				
		GMM-SVM (supervectors)		Total variability (i-vectors)		
SNR (dB)	Noises	w/o UP-AVR	With UP-AVR	w/o UP-AVR	With UP-AVR	DataBoost-UP
0	Car	22.01	38.28	32.93	41.41	54.43
	Factory	10.25	35.48	30.21	46.78	52.83
	Pink	22.80	45.18	37.26	51.86	57.94
	White	26.74	48.84	43.87	53.94	57.88
5	Car	25.81	61.98	54.11	66.59	73.33
	Factory	02.69	42.80	39.02	56.92	65.94
	Pink	18.98	47.72	43.83	57.47	65.97
	White	23.95	45.95	44.02	56.66	63.41

to the relatively moderate improvements in average EER, a significant reduction of average MinDCF value of 7.5×10^{-3} is noticed. This is comparatively much higher than the previously recorded average MinDCF reduction (see Table 5.11) of 3×10^{-3} due to the effect of UP-AVR.

Table 5.13 summarizes the relative EERs of the various SV systems developed in uniform background environments at 0 and 5 dB SNRs. In order to jointly represent the performances of the KL div and GUMI kernels, the mean of their relative EERs has been recorded under the GMM-SVM multicolumn. The significant performance improvements achieved in each stage of development of the i-vector based SV systems, can be more clearly deduced by the large relative EER metrics. A contrasting behavior in performance improvement of the i-vector based SV systems is observed at the two SNR levels. The colored noises (pink and white) which had comparatively lower performance accuracies, are the ones with higher relative EER improvements at 0 dB SNR. However, the environmental noises (car and factory) perform much better at 5 dB SNR. The average relative EERs (across both SNR levels) of the GMM-SVM based SV systems are 19.15% (without UP-AVR) and 45.78% (with UP-AVR), respectively. The corresponding EER values for the i-vector based SV systems are 40.66 and 53.96%, respectively. The DataBoost-UP algorithm in the i-vector framework outperforms the rest of the methods with an average relative EER of 61.47% across all environments.

5.5 Summary

This chapter explored the impact of robust speaker models for speaker verification in various noisy environments. Broadly, two types of hybrid modeling techniques (i.e., GMM-SVM and i-vectors SVM) were used to develop SV systems in uniform and varying background environments, respectively. The majority of studies were concentrated in the GMM-SVM based approach. Through extensive experimentation, it was established that robust SV performance could be achieved using GMM supervectors in a discriminative framework, in comparison to the traditional GMM-UBM framework. In particular, emphasis was laid on the significance of using partitioned utterances, for mitigating data imbalance, utterance-duration mismatch and small sample-size problems, respectively for improving performances in SVM based SV framework. In order to enhance SV performances in highly degraded environments, a low-dimensional channel robust representation of GMM supervectors (namely i-vectors), were alternatively used in a SVM framework. A novel boosting algorithm was proposed to address some inherent drawbacks in the standard utterance partitioning scheme and strengthening the SVM classification accuracy in highly degraded background environments.

References

1. D. Reynolds, T. Quatieri, R. Dunn, Speaker verification using adapted Gaussian mixture models. Digit. Signal Process. **10**(1), 19–41 (2000)
2. W. Campbell, J. Campbell, D. Reynolds, Support vector machines using GMM supervectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006)
3. W. Campbell, J. Campbell, D. Reynolds, E. Singer, P. Carrasquillo, Support vector machines for speaker and language recognition. Comput. Speech Lang. **20**, 210–229 (2006)
4. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006)
5. V. Wan, S.Renals, Speaker verification using sequence discriminant support vector machines. IEEE Trans. Acoust. Speech Audio Process. **13**(2), 203–210 (2005)
6. P. Kenny, G. Boulianne, P. Dumouchel, Eigenvoice modeling with sparse training data. IEEE Trans. Speech Audio Process. **13**(3), 345–354 (2005)
7. N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011)
8. S. Sarkar, K.S. Rao, Speaker verification in noisy environment using GMM supervectors, in *National Conference on Communications (NCC)*, Delhi (IIT Delhi, Delhi, 2013)
9. C.H. You, K.A. Lee, H. Li, An SVM kernel with GMM-Supervector based on the Bhattacharyya distance for speaker recognition. IEEE Signal Process. Lett. **16**(1), 49–52 (2009)
10. A. Solomonoff, C. Quillen, I. Boardman, Channel compensation for SVM speaker recognition, in *IEEE Workshop on Speaker and Language Recognition (Odyssey '04)*, Toledo, 2004, pp. 57–62
11. R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in *Proceedings of the 15th European Conference on Machine Learning*, Pisa, 2004, vol. 3201, pp. 39–50
12. Y. Tang, Y.Q. Zhang, N.V. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification. IEEE Trans. Syst. Man Cybern. Part B Cybern. **39**, 281–288 (2009)
13. J. Pelecanos, U. Chaudhari, G. Ramaswamy, Compensation of utterance length for speaker verification, in *ODYSSEY04 – The Speaker and Language Recognition Workshop*, Toledo, 2004
14. B. Fauve, N. Evans, J. Mason, Improving the performance of text-independent short duration SVM- and GMM-based speaker verification, in *Workshop on Speaker and Language Recognition (Odyssey)*, Stellenbosch, 2008
15. W. Rao, M.W. Mak, Boosting the performance of i-vector based speaker verification via utterance partitioning. IEEE Trans. Audio Speech Lang. Process. **21**(5), 1012–1022 (2013)
16. L.F. Chen, H.Y.M. Liao, M.T. Ko, J.C. Lin, G.J. Yu, A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognit. **33**(10), 1713–1726 (2000)

17. J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. J. Mach. Learn. Res. **6**, 483–502 (2005)
18. N. Sen, H. Patil, S.K.D. Mandal, K.S. Rao, Importance of utterance partitioning in SVM classifier with GMM supervectors for text-independent speaker verification, in *Mining Intelligence and Knowledge Exploration*. LNCS (Springer, Cham, 2013), pp. 780–789
19. N. Chawla, K. Bowyer, L. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 341–378 (2002)
20. N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Cavtat-Dubrovnik, 2003
21. P. Kang, S. Cho, EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems, in *ICONIP (1)*, Hong Kong, 2006, pp. 837–846
22. Z. Lin, Z. Hao, X. Yang, X. Liu, Several SVM ensemble methods integrated with undersampling for imbalanced data learning, in *Advanced Data Mining and Applications* (Springer, Berlin/Heidelberg, 2009), pp. 536–544
23. K. Veropoulos, C. Campbell, N. Cristianini, Contolling the sensitivity of support vector machines, in *Proceedings of International Joint Conference on Artificial Intelligence*, Stockholm, 1999
24. G. Wu, E. Chang, KBA: kernel boundary alignment considering imbalanced data distribution. IEEE Trans. Knowl. Data Eng. **17**(6), 786–795 (2005)
25. M.W. Mak, W. Rao, Utterance partitioning with acoustic vector resampling for GMM-SVM speaker verification. Speech Commun. **53**, 119–130 (2011)
26. S. Sarkar, K.S. Rao, Significance of utterance partitioning in GMM-SVM based speaker verification in varying background environment, in *16th International Oriental COCOSDA Conference*, Gurgoan, 2013
27. P. Kenny, G. Boulianne, P. Ouellet, P. Dumouchel, Speaker and session variability in GMM-based speaker verification. IEEE Trans. Audio Speech Lang. Process. **15**(4), 1448–1460 (2007)
28. P. Kenny, G. Boulianne, P. Ouellet, P. Dumouchel, Factor analysis simplified, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP '05)*, Philadelphia, 2005, vol. 1, pp. 637–640
29. N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, F. Castaldo, Support vector machines and joint factor analysis for speaker verification, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, Taipei, 2009, pp. 4237–4240
30. N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, P. Dumouchel, Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification, in *Proceeding of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH '09)*, Brighton, 2009
31. A.O. Hatch, S. Kajarekar, A. Stolcke, Within-class covariance normalization for SVM-based speaker recognition, in *Proceedings of the International Conference of Spoken Language Processing (ICSLP '05)*, Jeju, 2005
32. Y. Freund, R. Schapire, Experiments with a new boosting algorithm, in *Proceedings of Thirteenth International Conference on Machine Learning (ICML '96)*, Bari, 1996
33. A. Roy, M.M. Doss, S. Marcel, Boosted binary features for noise-robust speaker verification, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '10)*, Dallas, 2010, pp. 4442–4445
34. Y. Sun, S. Todorovic, J. Li, Reducing the overfitting of AdaBoost by controlling its data distribution skewness. Int. J. Pattern Recognit. Artif. Intell. **20**, 1093–1116 (2006)
35. H. Guo, H.L. Viktor, Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. ACM SIGKDD Explor. Newsl. Spec. Issue Learn. Imbalanced Datasets **6**, 30–39 (2004)