Utterance Partitioning with Acoustic Vector Resampling for GMM–SVM Speaker Verification

Man-Wai MAK and Wei RAO

Centre for Signal Processing,
Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University
enmwmak@polyu.edu.hk

Abstract

Recent research has demonstrated the merit of combining Gaussian mixture models and support-vector-machine (SVM) for text-independent speaker verification. However, one unaddressed issue in this GMM–SVM approach is the imbalance between the numbers of speaker-class utterances and impostor-class utterances available for training a speaker-dependent SVM. This paper proposes a resampling technique – namely utterance partitioning with acoustic vector resampling (UP-AVR) – to mitigate the data imbalance problem. Briefly, the sequence order of acoustic vectors in an enrollment utterance is first randomized, which is followed by partitioning the randomized sequence into a number of segments. Each of these segments is then used to produce a GMM supervector via MAP adaptation and mean vector concatenation. The randomization and partitioning processes are repeated several times to produce a sufficient number of speaker-class supervectors for training an SVM. Experimental evaluations based on the NIST 2002 and 2004 SRE suggest that UP-AVR can reduce the error rate of GMM–SVM systems.

Key words: Speaker verification; GMM-Supervectors (GSV); utterance partitioning, GMM–SVM; support vector machine; random resampling; data imbalance.

1 Introduction

The integration of Gaussian mixture models (GMMs) and support vector machines (SVMs) – namely GMM–SVM – has become one of the most promising approaches to text-independent speaker verification. This approach derives a GMM-supervector [1] by stacking the mean vectors of a MAP-adapted GMM

[2] that captures the acoustic characteristics of a speaker. The supervector is then presented to a speaker-dependent SVM for scoring. This SVM scoring approach is superior to the conventional likelihood-ratio scoring because the contribution of individual background speakers and the target speaker to the verification scores can be optimally weighted by the Lagrange multipliers of the target-speaker's SVM [3].

Nevertheless, a major drawback of the SVM scoring approach is that the number of target speaker utterances for training the target-speaker's SVM is very limited (typically only one enrollment utterance is available). Given that the number of background speakers' utterances is typically several hundreds, the limited number of enrollment utterances leads to a serious data imbalance problem. One problem of data imbalance is that the decision boundary of the resulting SVM will skew towards the minority (target speaker) class [4,5], causing high false-rejection rate unless the decision threshold is properly set to compensate for the bias. Another problem, as will be demonstrated in Section 3, is that the orientation of the decision boundary is largely dictated by the data in the majority (background speakers) class.

Because imbalanced classification occurs in many problem domains, a number of strategies have been proposed to alleviate the effect of imbalanced data on SVM classifiers. These strategies can be divided into two categories: data processing approaches and algorithmic approaches. The former attempts to rebalance the training data without changing the SVM training algorithm. This category can be further divided into (1) over-sampling [6,7] where more positive (minority class) training examples are generated from existing data, (2) under-sampling [8,9] where a subset of negative (majority class) training samples are selected for each entity in an ensemble of SVMs, and (3) combinations of over- and under-sampling [5]. While studies have shown that the data processing approaches can improve the performance of SVMs in some situations, they do have their own problems. For example, over-sampling will increase the number of support vectors, causing computational burden for large datasets. Some over-sampling techniques (e.g. SMOTE [6]) assume that the samples on the line joining two neighboring positive samples are also positive. This assumption may be invalid in some situations. Although under-sampling can help move the decision boundary towards the majority class, it causes information loss if useful samples are discarded. A recent study [10] also suggests that for some applications, the performance of SVMs with over- or under-sampling could be poorer than those without any sampling.

The algorithmic approaches attempt to modify the training algorithms to mitigate the effect caused by data imbalance. One earlier attempt is to assign different misclassification cost to positive and negative training samples [11,12]. However, studies [4] have shown that this approach is not very effective, because increasing the value of the Lagrange multipliers of the minority

class (due to the increase in the penalty factor) will also increase some of the Lagrange multipliers in the majority class to satisfy the constraint $\sum_i \alpha_i y_i = 0$. Another algorithmic approach is to modify the kernel according to the distribution of training data [4]. This approach, however, requires longer classification time than the standard SVM.

Unlike many other problem domains, the data imbalance problem in GMM–SVM speaker verification is special in that the number of minority-class samples (enrollment utterances) is extremely small. In fact, it is not uncommon to have only one enrollment utterance per client speaker. This extreme data imbalance excludes the use of over-sampling methods such as SMOTE where minority-class samples are generated based on the existence of some (but not one) minority-class samples. Under-sampling is also not an option, because of the information loss that arises from discarding important background speakers.

In this paper, we look at over-sampling in another dimension. Instead of creating more minority-class samples from existing ones, we generate minority-class samples by partitioning the sequence of acoustic vectors in the enrollment utterance into a number of segments or sub-utterances, with each segment producing one GMM-supervector. To increase the number of segments, one may reduce the length of sub-utterances. However, this will inevitably compromise the representation power of the sub-utterances. Here, we propose to address this issue by randomizing the sequence order before partitioning takes place. This randomization and partitioning process can be repeated several times to produce a desirable number of GMM-supervectors. More precisely, if the process is repeated R times and for each time the sequence is divided into N segments, a total of RN GMM-supervectors will be generated. In each repetition, N GMM-supervectors are generated from a different set of acoustic vectors. The randomization process ensures that the GMM-supervectors are different from repetition to repetition. However, as the number of acoustic vectors in an utterance is finite, a large R will inevitably increase the correlation among the GMM-supervectors. In this work, R and N were found empirically.

The paper is organized as follows. Section 2 introduces the concept of GMM–UBM and GMM–SVM. Section 3 explains why the limited number of enrollment utterances per target speaker can cause problems in GMM–SVM speaker verification and proposes using utterance partitioning with acoustic vector resampling to mitigate the problem. In Sections 4 and 5, we report our evaluations based on NIST 2002 and 2004 SRE, which show that the proposed utterance partitioning approach can reduce the EER and minimum DCF of GMM–SVM systems. Concluding remarks and future work are then given in Section 6.

2 GMM–UBM and GMM–SVM

2.1 GMM–UBM

The idea of GMM–UBM is to create a target-speaker's Gaussian mixture model (GMM) via maximum *a posteriori* (MAP) adaptation of a universal background model (UBM) [2]. Specifically, given an enrollment utterance with acoustic vector sequence $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, the following formulae are applied to the mean vectors $\boldsymbol{\mu}_i$ of the UBM to obtain the adapted mean vectors $\hat{\boldsymbol{\mu}}_i$:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_i &= \alpha_i E_i(X) + (1-\alpha_i)\boldsymbol{\mu}_i, \quad i = 1, \ldots, M, \\
\alpha_i &= \frac{n_i(X)}{n_i(X) + r} \\
n_i(X) &= \sum_{t=1}^{T} \Pr(i|\mathbf{x}_t) \\
E_i(X) &= \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|\mathbf{x}_t)\mathbf{x}_t, \\
\Pr(i|\mathbf{x}_t) &= \frac{\lambda_i p_i(\mathbf{x}_t)}{\sum_{j=1}^{M} \lambda_j p_j(\mathbf{x}_t)}
\end{aligned} \tag{1}$$

where λ_i and $p_i(\mathbf{x})$ are the mixture weight and density function of the i-th mixture, respectively, and r is a relevance factor controlling the degree of adaptation.

During verification, the verification score of a claimant utterance, $\text{utt}^{(c)}$, is obtained by computing the ratio between the target-speaker likelihood and background-speaker likelihood, i.e.,

$$S_{\text{GMM–UBM}}(\text{utt}^{(c)}) = \log p(X^{(c)}|\Lambda^{(s)}) - \log p(X^{(c)}|\Lambda^{(b)}), \tag{2}$$

where $X^{(c)}$ is the sequence of acoustic vectors (typically MFCCs [13] and their derivatives) derived from $\text{utt}^{(c)}$, and $\Lambda^{(s)}$ and $\Lambda^{(b)}$ are the GMMs representing the target speaker and background speakers, respectively. Because the parameters of the two likelihood functions are estimated separately, the scoring function in Eq. 2 does not make full use of the discriminative information in the training data [14]. [1]

[1] When estimating the parameters of a target-speaker model in GMM–UBM systems, we do not strike to maximize the discrimination between the target-speaker's speech from impostors' speech as in discriminative training. Although a target-speaker model is adapted from the UBM, they are not computed "jointly". This concept has been explained in [14].

2.2 GMM–SVM

The idea of GMM–SVM [1] is to harness the discriminative information embedded in the training data by constructing an SVM that optimally separates the GMM of a target speaker from the GMMs of background speakers. Like the GMM–UBM approach, a speaker-dependent GMM is created by adapting from the UBM via MAP adaptation [2]. However, unlike GMM–UBM, the mean vectors of the speaker-dependent GMM are stacked to form a GMM-supervector. This target supervector together with the supervectors corresponding to individual background speakers are used to train a target-speaker SVM. Therefore, in addition to a GMM, each target speaker is also represented by an SVM that operates in a space (called GMM-supervector space) with axes corresponding to individual coefficients of GMM mean vectors.

In GMM–SVM, given the SVM of target speaker s, the verification score of $\text{utt}^{(c)}$ is given by

$$S_{\text{GMM–SVM}}(\text{utt}^{(c)}) = \alpha_0^{(s)} K\left(\text{utt}^{(c)}, \text{utt}^{(s)}\right) - \sum_{i \in \mathcal{S}^{(b)}} \alpha_i^{(s)} K\left(\text{utt}^{(c)}, \text{utt}^{(b_i)}\right) + d^{(s)}, \tag{3}$$

where $\alpha_0^{(s)}$ is the Lagrange multiplier corresponding to the target speaker,[2] $\alpha_i^{(s)}$'s are Lagrange multipliers corresponding to the background speakers, $\mathcal{S}^{(b)}$ is a set containing the indexes of the support vectors in the background-speaker set, and $\text{utt}^{(b_i)}$ is the utterance of the i-th background speaker. Note that only those background speakers with non-zero Lagrange multipliers have contribution to the score. The kernel function $K(\cdot, \cdot)$ can be of many forms. The most common being the Mahalanobis kernel (also called GMM-supervector kernel) [1]:

$$K\left(\text{utt}^{(c)}, \text{utt}^{(s)}\right) = \sum_{j=1}^{M} \left(\sqrt{\lambda_j} \boldsymbol{\Sigma}_j^{-\frac{1}{2}} \boldsymbol{\mu}_j^{(c)}\right)^{\mathrm{T}} \left(\sqrt{\lambda_j} \boldsymbol{\Sigma}_j^{-\frac{1}{2}} \boldsymbol{\mu}_j^{(s)}\right) \tag{4}$$

where λ_j and $\boldsymbol{\Sigma}_j$ are the mixture weights and covariances of the UBM, respectively, and $\boldsymbol{\mu}_j^{(c)}$ and $\boldsymbol{\mu}_j^{(s)}$ are the j-th mean vector of the GMM belonging to claimant c and speaker s, respectively.

Eq. 4 can be written in a more compact form

$$K\left(\text{utt}^{(c)}, \text{utt}^{(s)}\right) = \left\langle \boldsymbol{\Omega}^{-\frac{1}{2}} \overrightarrow{\boldsymbol{\mu}}^{(c)}, \boldsymbol{\Omega}^{-\frac{1}{2}} \overrightarrow{\boldsymbol{\mu}}^{(s)} \right\rangle \tag{5}$$

where

$$\boldsymbol{\Omega} = \text{diag}\left\{\lambda_1^{-1}\boldsymbol{\Sigma}_1, \ldots, \lambda_M^{-1}\boldsymbol{\Sigma}_M\right\} \quad \text{and} \quad \overrightarrow{\boldsymbol{\mu}} = \left[\boldsymbol{\mu}_1^{\mathrm{T}}, \ldots, \boldsymbol{\mu}_M^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{6}$$

[2] We assume one enrollment utterance per target speaker, which is the case in NIST SRE 2002.

In practice, $\mathbf{\Sigma}_j$'s are assumed to be diagonal.

3 Utterance Partitioning for GMM–SVM

In GMM–SVM systems, it is not uncommon to have only one speaker-class's supervector for training. The problem is that the SVM's decision boundary is largely governed by the impostor-class supervectors (support vectors). This situation is illustrated in Fig. 1(a). There is a region in the feature space where the positive-class's support vector (encircled $\square$) can move around without affecting the orientation of the decision boundary, but a small change in the negative-class' support vectors (encircled $*$) can tilt the decision boundary.

To increase the influence of speaker-class data on the decision boundary, one may use more enrollment utterances, which means more supervectors from the speaker class. However, as mentioned earlier, it is not practical to request users to provide multiple enrollment utterances. To solve this problem without introducing extra burden on users, this paper proposes partitioning an enrollment utterance into a number of sub-utterances. The method is referred to as *utterance partitioning* (UP). Given an enrollment utterance, a large number of partitions will produce many sub-utterances, but their length may be too short to represent the speaker. On the other hand, if the number of partitions is too small, the benefit of utterance partitioning diminishes. Obviously, there is a trade-off between the length of the sub-utterances and the representation capability of the resulting GMM supervectors.

One possible way to generate more sub-utterances with reasonable length is to use the notion of random resampling in bootstrapping [15]. The idea is based on the fact that the MAP adaptation algorithm uses the statistics of the whole utterance to update the GMM parameters (see Eq. 1). In other words, changing the order of acoustic vectors will not affect the resulting MAP-adapted model. Therefore, we may randomly rearrange the acoustic vectors in an utterance and then partition the utterance into N sub-utterances and repeat the process as many times as appropriate. More precisely, if this process is repeated R times, we obtain RN sub-utterances from a single enrollment utterance. We refer to this approach as utterance partitioning with acoustic vector resampling (UP-AVR). Its procedure is as follows:

Step 1: For each utterance from the background speakers, divide the utterance into N partitions (sub-utterances) and compute their acoustic vectors (MFCCs and their derivatives).

Step 2: For each utterance from the background speakers, compute its acoustic vectors (MFCCs and their derivatives) and divide the vectors into N partitions (sub-utterances).

Step 3: For each background speaker, use his/her N sub-utterances and full-length utterance to create $N+1$ background GMM-supervectors. For B background speakers, this procedure results in $B(N+1)$ background supervectors.

Step 4: Given an enrollment utterance of a target speaker, compute its acoustic vectors and randomize their sequence of occurrences in the utterance. Divide the randomized sequence of acoustic vectors into N partitions (sub-sequences). Use the N sub-sequences to create N GMM-supervectors by adapting the UBM.

Step 5: Repeat Step 3 R times to obtain RN target speaker's supervectors; together with the full-length utterance, form $RN+1$ speaker's supervectors.

Step 6: Use the $RN+1$ supervectors created in Steps 3 and 4 as positive-class data and the $B(N+1)$ background supervectors created in Step 2 as negative-class data to train a linear SVM for the corresponding target speaker.

Figure 2 illustrates the procedure of UP-AVR. The same partitioning strategy are applied to both target-speaker utterances and background utterances so that the length of target-speaker's sub-utterances matches that of the background speakers' sub-utterances. Matching the duration of target-speaker utterances with that of background utterances has been found useful in previous studies [16].

The advantages of the utterance partitioning approach are two-fold. First, it can increase the influence of positive-class data on the decision boundary. Second, when the original enrollment utterances are significantly longer than the verification utterances, utterance partitioning can create sub-utterances with length that matches the verification utterances. This can reduce the mismatches between the test supervectors and the enrollment supervectors, because the amount of MAP adaptation depends on the length of the adaptation utterances.

4 Experiments

4.1 Speech Data, Features, and Scoring

NIST 1999–2002 Speaker Recognition Evaluation (SRE), NIST 2004 SRE, [3] and Fisher [17] were used in the experiments. NIST'99–01 SRE and Fisher were used as development data, and NIST'02 and NIST'04 were used for per-

[3] http://www.itl.nist.gov/iad/mig/tests/sre

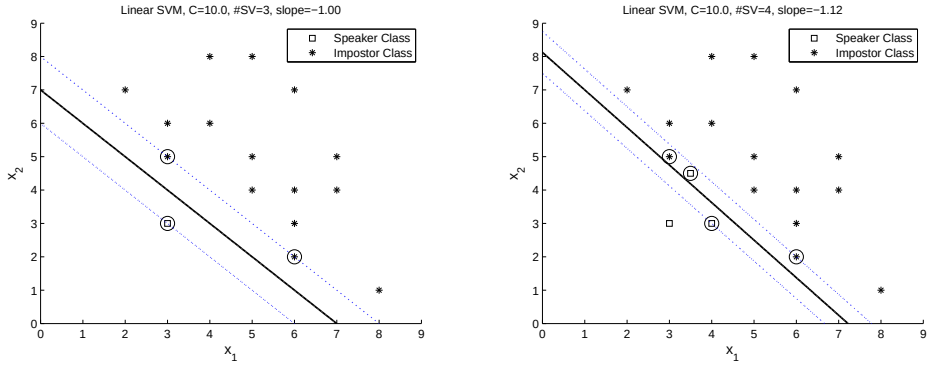


Fig. 1. A two-class problem illustrating the imbalance between the number of positive-class samples and negative-class samples. (a) The orientation (slope) of the decision boundary depends largely on the negative-class data. (b) Adding more positive-class data can enhance the influence of the positive-class data on the decision boundary (slope changes from −1.0 to −1.12).

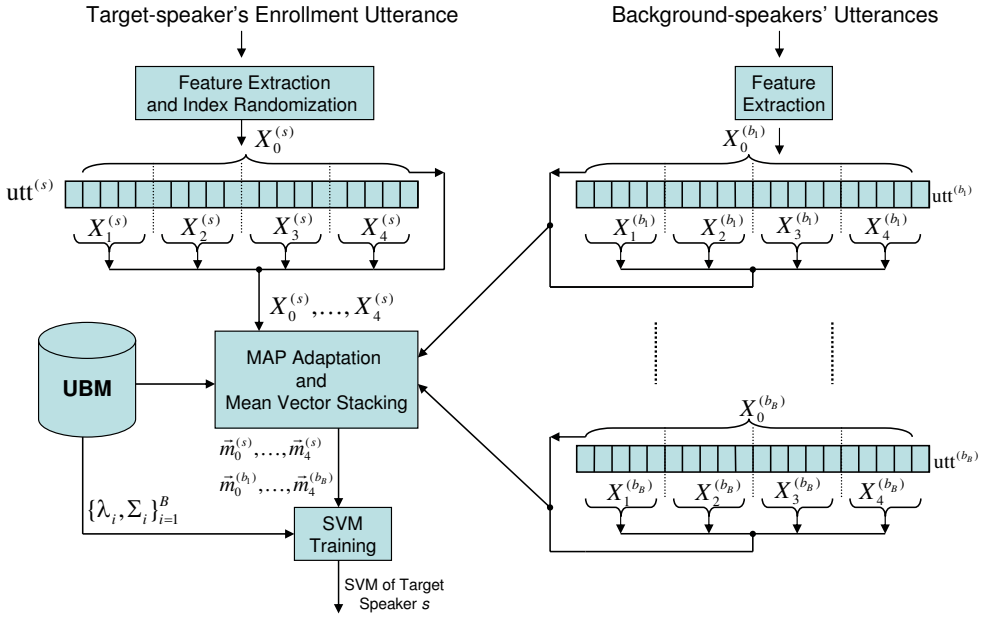


Fig. 2. The procedure of utterance partitioning with acoustic vector resampling (UP-AVR). Note that randomization and creation of target-speaker's supervectors can be repeated several times to obtain a sufficient number of target-speaker's supervectors.

formance evaluations.[4] Table 1 summarizes the roles played by these corpora in the evaluations.

NIST'02 contains cellular phone conversations of 139 male and 191 female

[4] Hereafter, all NIST SREs are abbreviated as NIST'XX, where XX stands for the year of evaluation.

target speakers taken from Switchboard Cellular Part 2. Each target speaker provides 2 minutes of speech for training. There are 2,983 true-speaker trials and 36,287 impostor attempts.

NIST'04 contains 28 evaluation conditions. This paper focuses on the 1side-1side condition, i.e., each of the training and test segments contains a whole conversation side. This condition contains 246 male and 376 female target speakers, each providing 5 minutes of speech (including silence) for training. The evaluation condition also contains 2,386 true-speaker trials and 23,838 impostor attempts, with each test segment containing 5 minutes of speech (including silence).

NIST'01 contains 2,350 cellular phone conversations extracted from the Switchboard-II Phase IV Corpus. All of these utterances were used for training the gender-dependent background models in NIST'02 evaluation. All of the utterances from the training sessions in the corpus were used as gender-dependent impostor data for training the target-speaker SVMs. Test utterances with length (after silence removal) longer than 25 seconds were used for creating the T-norm [18] speaker models, which amount to 127 male and 145 female speaker models for T-norm. The corpus was also used for computing the projection matrices in Nuisance Attribute Projection (NAP) [19]. Specifically, speakers with multiple conversations were identified and the conversations of these speakers are assumed to be extracted from different sessions. This amounts to 74 male speakers and 100 female speakers, each providing 12 conversations on average. The number of nuisance dimensions (corank in [20]) to be projected out is eight for male and one for female. These numbers were found empirically to produce the best performance on a baseline system (see Section 5.2).

For the NIST'04 evaluation, the Fisher corpus was used for training the gender-dependent UBMs. A subset of speakers was used for training the gender-dependent T-norm models, and another subset was used as impostor-class data for training the target-speaker SVMs and T-norm SVMs. Finally, 236 male and 266 female speakers from NIST'99 and NIST00 were used for estimating the gender-dependent NAP matrices. Each of these speakers has at least 8 utterances. The NAP corank was set to 64 for both genders.

For each utterance, an energy-based voice activity detector was used to remove the silence regions. Twelfth-order MFCCs [13] plus their first derivative were extracted from the speech regions of the utterance, leading to 24-dim acoustic vectors. Cepstral mean normalization [21] was applied to the MFCCs, followed by feature warping [22]. Then, UP-AVR was applied to the feature vectors of each utterance.

For GMM–UBM, T-norm [18] or ZT-norm (Z-norm followed by T-norm) was applied during the scoring stage. For GMM–SVM, NAP was applied to all

	UBMs	T-norm Models	Impostor-class of SVMs	NAP Matrices
NIST'02 Eval	NIST'01: 1006 male and 1344 female utterances	NIST'01: 127 male and 145 female speakers from test sessions	NIST'01: 112 male and 122 female speakers from training sessions	NIST'01: 74 male and 100 female speakers from test sessions#
NIST'04 Eval	Fisher: 1100 male and 1640 female utterances	Fisher: 200 male and 200 female speakers	Fisher: 300 male and 300 female speakers	NIST'99 and NIST'00: 236 male and 266 female speakers from test sessions#

Table 1
The roles played by different corpora in the performance evaluations. #Only speakers with 8 or more utterances were used for estimating the NAP matrices.

GMM-supervectors, followed by T-norm scoring. Unlike GMM–UBM, we did not observe any performance advantage of ZT-norm over T-norm on the baseline GMM–SVM system. Therefore, we only report the results of T-norm in this paper.

4.2 Speaker Models

The classical GMM–UBM [2] and GMM–SVM [1] were used as the baselines for comparison. For the GMM–UBM systems, the number of mixtures for gender-dependent UBMs is 1,024. The GMMs of target speakers were adapted from the UBMs using MAP adaptation [2] with relevance factor r in Eq. 1 set to 16. For the GMM–SVM systems, the number of mixtures was set to 256 for most of the experiments because this model size achieves the best performance in the baseline GMM–SVM system (see Table 2). Each supervector in the GMM–SVM comprises the means of a MAP-adapted GMM. For each target-speaker SVM, positive (target-speaker) class supervectors were obtained by stacking the means of the MAP-adapted GMMs created from the utterance of the corresponding speaker, whereas the negative (impostor) class supervectors were obtained from the either NIST'01 or Fisher (see Table 1). SVM^{light} [23] was used for training the SVMs.

5 Results and Discussions

5.1 Statistical Properties of GMM-Supervectors

The feature dimension of GMM-supervectors is MD, where M is the number of mixtures in a GMM and D is the dimension of acoustic vectors. For systems that use a large number of mixtures (e.g., $M \geq 1024$), the dimension of the GMM-supervectors will become very large, which introduces excessive computational burden on the training of speaker-dependent SVMs. If M is too small, the resulting supervectors may not be able to represent the characteristics of the target speakers. Nevertheless, one may ask: Among all the features in the supervectors, how many of them are relevant or useful for recognizing speakers?. To answer this question, we computed the variances of MD-dimensional features from 50 normalized GMM-supervectors $\mathbf{\Omega}^{-\frac{1}{2}}\overrightarrow{\boldsymbol{\mu}}^{(b_k)}$, where M ranges from 64 to 1,024, $D = 24$, $k = 1, \ldots, 50$, and $\mathbf{\Omega}$ and $\overrightarrow{\boldsymbol{\mu}}$ are defined in Eq. 6. Features were then sorted in descending order of variances. Fig. 3 shows the variances of features against the feature indexes. The horizontal line is a threshold (0.05) below which the features are considered to have no significant contribution to the classification task. Evidently, when $M \geq 512$, a large percentage of features have variances below the threshold, which means that the resulting SVMs have input dimension larger than necessary. A model size of 256 mixtures seems to be a good compromise. This observation also agrees with the verification performance shown in Table 2, where the best speaker verification performance (in terms of equal error rate and minimum decision cost (DCF) [24]) is obtained when $M = 256$. Based on this finding, we set $M = 256$ for the rest of the experiments.

5.2 Effect of Varying the Corank in NAP

Fig. 4 shows the verification performance of GMM–SVM systems using different NAP coranks (nuisance dimension [20]) under the core test condition in NIST'02 and NIST'04. The results suggest a small corank is appropriate for NIST'02 whereas NIST'04 requires a larger corank. We conjecture that the reason of using a smaller corank for NIST'02 is that the session variation in NIST'02 is smaller than that in NIST'04. Although many speakers in NIST'02 have participated in more than one recording session, they used the same cell-phone model, network operator (e.g. Bell Atlantic), and transmission type (e.g., CDMA) in different sessions. [5] Therefore, session variation is mainly due to the variation in intra-speaker characteristics rather than varia-

[5] This information can be found in the file 'sid02_1sp.v2.ref' in NIST'02.

tion in channel characteristics. On the other hand, in NIST'04, speakers may use different handsets in different sessions, causing larger session variation.

To verify this conjecture, we computed the eigenvalues of NAP covariance matrices (Eq. 12 in [19] but normalized by the number of column vectors in $\mathbf{A}$) for different corpora and plotted the results in Fig 5. Evidently, NIST'02 has the smallest eigenvalues among all speech corpora. Therefore, its session variation is also the smallest.

Comparisons between the performance at Corank = 0 (i.e., without NAP) and that at the best corank in Fig. 4 suggest that NAP is more effective in NIST'04 than in NIST'02. This evidence further supports our conjecture that session variation in NIST'02 is smaller than that in NIST'04. Bear in mind that NAP is designed to alleviate the effect of session variation. If session variation is small, it is reasonable that NAP does not have significant effect on verification performance. In fact, in the original paper of NAP, Solomonoff et al. [25] also found that NAP is effective in cross-channel (carbon-button–electret) scenarios but minor degradation is seen for same channel conditions.

Fig. 4 also suggests that the optimal corank is gender-dependent. To find out the reason of this result, we plot the eigenvalues of gender-dependent NAP covariance matrices of NIST'02 and NIST'04 in Fig. 6. Evidently, the eigenvalues of female speakers are smaller than the male counterpart, suggesting that the GMM-supervectors of female speakers in these corpora have smaller variations. Bear in mind that if there is no intra-speaker variation, the optimal corank should be zero, meaning that the projection matrix should be an identity matrix. A small eigenvalue suggests that a small corank should be used, confirming the results in Fig. 4 where the optimal corank for female speakers is smaller than that of male speakers, especially in NIST'02.

5.3 Effect of Varying the Number of Speaker-Class Supervectors

Table 3 shows the effect of varying the number of target-speaker's GMM-supervectors (GSVs) on the verification performance in NIST'02. The GSVs were obtained by either UP or UP-AVR. In all cases, the same partitioning strategy was applied to both target-speaker utterances and background-speaker utterances so that the length of target-speaker sub-utterances matches that of the background-speaker sub-utterances (see Fig. 2). Because UP-AVR randomizes the feature indexes before partitioning the utterances, the supervectors created will be different from simulation run to simulation run. To investigate the reliability of the estimated EER and minimum DCF, 17 independent simulation runs were performed. The mean and standard deviation of 17 EERs and minimum DCFs are shown in the last row of Table 3.

Table 3 shows that for the same number of target-speaker GSVs, UP-AVR achieves a lower EER than that of UP. Although there is a slight increase in minimum DCF, except for the 2nd row, the increase in minimum DCF is not as significant as the decrease in EERs. Table 3 also shows that setting $N = 32$ for UP leads to very poor performance. The reason is that excessive partitioning will produce very short sub-utterances, making the resulting speaker-class GSVs almost identical to the GSV of the UBM after MAP adaptation.

Figures 7(a) and (b) show the trend of EER and minimum DCF when the number of speaker-class supervector increases. The figures demonstrate that utterance partitioning can reduce EER and minimum DCF. More importantly, the most significant performance gain is obtained when the number of speaker-class supervectors increases from 1 to 5, and the performance levels off when more supervectors are added. This is reasonable because a large number of positive supervectors will only result in a large number of zero Lagrange multipliers for the speaker class and increase the correlation among the synthesized supervectors.[6] Fig. 7(c) shows the p-values of McNemar's tests [26] on the pairwise differences between the EERs under different numbers of speaker-class supervectors. The first row suggests that increasing the number of speaker-class supervectors from 1 to 5 and beyond by means of UP-AVR can bring significant reduction in EER. On the other hand, five speaker-class supervectors may already be sufficient because further increase in this number does not bring significant performance gain, as evident by the high p-values in the entries other than the first row.

Fig. 8 shows the EERs of UP-AVR for different numbers of partitions (N) and resampling (R); when $R = 0$, UP-AVR is reduced to UP. Evidently, for small number of partitions (e.g., $N = 2$ and $N = 4$), UP-AVR ($R \geq 1$) performs better than UP ($R = 0$), suggesting that resampling can help create better GMM–SVM speaker models. However, when the number of partitions increases (e.g, $N = 8$), the advantage of resampling diminishes. This result agrees with our earlier argument in Section 3 that when the number of partitions is too large, the length of sub-utterances will become too short, causing their corresponding supervectors almost identical to that of the UBM.

Table 5 shows the performance of UP-AVR in NIST'04 when the number of speaker-class supervectors increases from 5 to 201. The results suggest that with just 5 speaker-class supervectors (UP-AVR(5)), significant reduction in EER can be obtained. However, adding extra speaker-class supervectors can only reduce the EER slightly, which again confirms our earlier argument that it is not necessary to generate excessive number of speaker-class supervectors.

[6] Our preliminary investigation on several speakers suggest that when the number of speaker-class supervectors is greater than 40, about half of the supervectors are not support vectors.

The p-value of McNemar's test [26] between System E and System F in Table 5 is 7×10^{-9}. Because the p-value is significantly smaller than 0.005 and the EER of System F is higher than other systems that use UP-AVR, we conclude that all of the systems that use UP-AVR are significantly better than the one without using UP-AVR.

5.4 GMM–UBM Versus GMM–SVM with UP-AVR

Table 4 shows the EER and minimum DCF of the best performing systems in NIST'02 under different configurations. The results clearly demonstrate the merit of utterance partitioning, particularly the one with acoustic vector resampling. The EER (8.16%) and min DCF (0.0337) achieved by our system are either comparable or better than other published results in the literature, including [27,3,28].

Table 5 shows the performance of GMM–UBM and GMM–SVM with UP-AVR in NIST'04. Again, the results demonstrate the merit of UP-AVR and the insensitivity of the algorithm with respect to the number of target-speaker supervectors. The lowest EER and minimum DCF are also lower than those published in the literature (e.g., [29–33]). The EER of the GMM–UBM system is significantly higher than that of the GMM–SVM systems, although it is comparable to that of others in the literature (e.g., [29]). Further work is required to improve the performance of the GMM–UBM system, e.g., using eigenchannel [34].

Fig. 10 plots the minimum DCF against the EER for various configurations. It highlights the amount of performance gain that can be obtained by UP-AVR. Fig. 9 shows the DET curves of various systems, which suggest that GMM–SVM with utterance partitioning is significantly better than the baseline GMM–SVM and GMM–UBM systems for a wide range of decision thresholds.

Our results and other published results in the literature suggest that the EER and minimum DCF in NIST'02 and NIST'04 are higher than those achievable in more recent corpora such as NIST'08. The reason may be that recent results on NIST'08 are typically based on joint factor analysis using a large amount of background data to train the Eigenchannel and Eigenvoice matrices. For example, Dehak et al. [35] used Switchboard, NIST'04, and NIST'05 to estimate these matrices and achieved an EER of 6.55% (all trials). As the amount of data prior to NIST'04 is significantly less than the amount of data prior to NIST'08, it will be difficult to reduce the EER of NIST'04 to a level comparable to that of NIST'08.

Performance	No. of Centers in GSV				
	64	128	256	512	1024
EER (%)	10.85	9.84	**9.05**	9.98	11.78
Minimum DCF	0.0414	0.0372	**0.0362**	0.0367	0.0428

Table 2
The effect of varying the number of Gaussian components in GMM-supervectors on the verification performance of the GMM–SVM baseline system (with NAP and T-norm) in NIST'02.

6 Conclusion

An approach to increase the number of target-speaker's supervectors in GMM–SVM speaker verification has been proposed. By randomizing the sequence order of acoustic vectors in an enrollment utterance, a useful set of speaker-class supervectors can be generated. Evaluations based on NIST 2002 SRE and NIST 2004 SRE show that the generated supervectors can alleviate the data imbalance problem and help the SVM learning algorithm to find better decision boundaries, thereby improving the verification performance. The proposed resampling technique has important implications to practical implementation of speaker verification systems because it reduces the number of enrollment utterances and thereby reducing the burden and time users spent on speech recording, which is one of the major obstacles in the commercialization of speaker verification technologies.

7 Acknowledgements

This work was in part supported by Center for Signal Processing, The Hong Polytechnic University (1-BB9W) and Research Grant Council of The Hong Kong SAR (PolyU 5264/09E).

References

[1] W. M. Campbell, D. E. Sturim, D. A. Reynolds, Support vector machines using GMM supervectors for speaker verification, IEEE Signal Processing Letters 13 (2006) 308–311.

[2] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (2000) 19–41.

No. of Target-Speaker's GSVs	GSV Generation Method	EER (%)	Min. DCF × 100
1	None ($N = 1$)	9.05 [8.74,9.27]	3.61 [3.45,3.77]
3	UP ($N = 2$)	8.66 [8.36,8.83]	3.45 [3.30,3.59]
	UP-AVR ($N = 4$, $R = 1$)	8.43 [8.09,8.55]	3.65 [3.49,3.81]
5	UP ($N = 4$)	8.46 [8.18,8.64]	3.42 [3.27,3.58]
	UP-AVR ($N = 4$, $R = 1$)	8.21 [7.98,8.46]	3.43 [3.27,3.59]
9	UP ($N = 8$)	8.30 [8.04,8.53]	3.36 [3.21,3.52]
	UP-AVR ($N = 4$, $R = 2$)	8.18 [7.80,8.25]	3.38 [3.22,3.52]
33	UP ($N = 32$)	15.06 [14.63,15.23]	5.14 [4.96,5.29]
	UP-AVR ($N = 4$, $R = 8$)	8.16 [7.93,8.41]	3.38 [3.23,3.53]

Table 3
Effect of varying the number of speaker-class supervectors on speaker verification performance in NIST'02. The speaker-class supervectors were generated by utterance partitioning (UP) and utterance partitioning with acoustic vector resampling (UP-AVR). The first column shows the number of speaker-class GSVs, which include the GSVs created by UP or UP-AVR and the GSV produced by the full-length utterance. "None" in the 2nd column means a full-length utterance was used to produce a single supervector for training a speaker-dependent SVM. N and R are the number of partitions per full-length utterance and the number of times resampling were performed to obtain the speaker-class GSVs. When the number of GSVs generated by UP-AVR is larger than the number of required speaker's GSV (e.g., 2nd row with UP-AVR, $N = 4$ and $R = 1$), the speaker's GSVs were randomly selected from the pool. The same utterance partitioning procedure was also applied to the background speakers' utterances so that the length of partitioned background utterances matches with that of the speaker's partitioned utterances. The number of background GSVs is $B(N+1)$, where B is the number of background speakers (112 for male and 122 for female). The numbers inside the square brackets are the 90% confidence intervals of FAR (at equal error threshold) and minimum DCF found by bootstrapping techniques [36,37].

[3] S. X. Zhang, M. W. Mak, Optimization of discriminative kernels in SVM speaker verification, in: Interspeech'09, Brighton, 2009, pp. 1275–1278.

[4] G. Wu, E. Y. Chang, KBA: Kernel boundary alignment considering imbalanced data distribution, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 786–795.

[5] Y. Tang, Y. Zhang, N. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification, IEEE Trans. on System, Man, and Cybernetics, Part B 39 (1) (2009) 281–288.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Artificial Intelligence and Research 16 (2002) 321V357.

[7] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, in: Proc. of the 7th

Verification Method	EER (%)	Minimum DCF
(A) GMM–UBM	11.19	0.0546
(B) GMM–UBM+TNorm	10.29	0.0428
(C) GMM–UBM+ZTNorm	9.39	0.0393
(D) GMM–SVM+NAP+TNorm	9.05	0.0362
(E) GMM–SVM+NAP+TNorm+UP-AVR(33)	**8.16**	**0.0337**

Table 4
Performance of GMM–UBM, GMM–SVM, and GMM–SVM with utterance partitioning in NIST'02. The numbers inside the parentheses indicate the number of speaker-class supervectors used for training a speaker-dependent SVM, which include the supervectors generated by UP-AVR ($N = 4$, $R = 8$) and the full-length utterance.

Method	EER	MinDCF
(A) GMM–UBM	17.05	0.0615
(B) GMM–UBM+TNorm	16.05	0.0601
(C) GMM–UBM+ZTNorm	15.95	0.0638
(D) GMM–SVM+TNorm	13.40	0.0516
(E) GMM–SVM+NAP+TNorm	10.42	0.0458
(F) GMM–SVM+NAP+TNorm+UP-AVR(5)	9.67	0.0421
(G) GMM–SVM+NAP+TNorm+UP-AVR(33)	9.63	0.0424
(H) GMM–SVM+NAP+TNorm+UP-AVR(61)	9.63	0.0422
(I) GMM–SVM+NAP+TNorm+UP-AVR(81)	9.57	0.0422
(J) GMM–SVM+NAP+TNorm+UP-AVR(93)	9.57	0.0424
(K) GMM–SVM+NAP+TNorm+UP-AVR(101)	**9.46**	**0.0419**
(L) GMM–SVM+NAP+TNorm+UP-AVR(201)	9.58	0.0421

Table 5
Performance of GMM–UBM, GMM–SVM, and GMM–SVM with utterance partitioning in NIST'04 (core test, all trials). The numbers inside the parentheses indicate the number of speaker-class supervectors used for training a speaker-dependent SVM, which include the supervectors generated by UP-AVR ($N = 4$, $R = 1, 8, 15, 20, 23, 25, 50$) and the full-length utterance.

European Conference on Principles and Practice of Knowledge Discovery in Databases, 2003, p. 107V119.

[8] P. Kang, S. Cho, EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems, in: I. King, et al. (Ed.), ICONIP'06, Vol. LNCS 4232, 2006, pp. 837–846.

[9] Z. Y. Lin, Z. F. Hao, X. W. Yang, X. L. Liu, Several SVM ensemble methods integrated with under-sampling for imbalanced data learning, in: Advanced Data Mining and Applications, Vol. 5678/2009 of LNCS, Springer, 2009, pp. 536–544.

[10] A. Sun, E. P. Lim, Y. Liu, On strategies for imbalanced text classification using

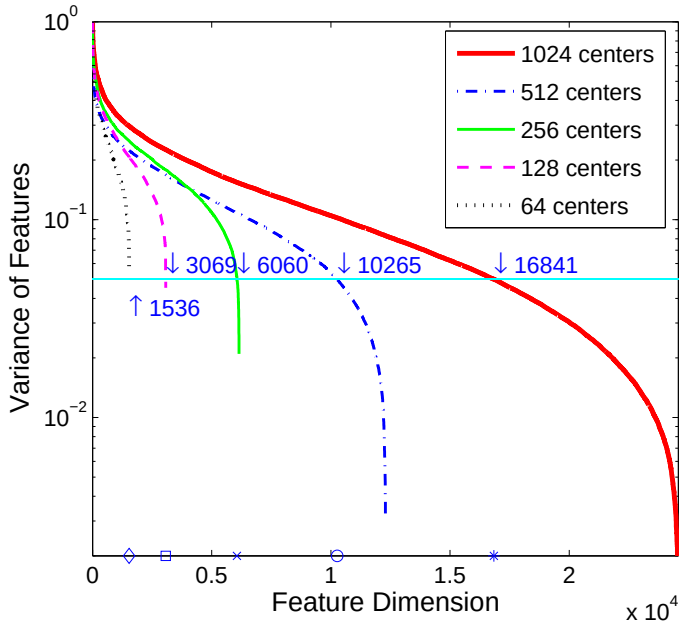


Fig. 3. The variance of features in the GMM-supervectors with different numbers of mixture components. To facilitate comparison, the maximum variance has been normalized to 1.0. The horizontal line is the variance threshold (= 0.05) above which the features are deemed relevant. The numbers along the horizontal line are the number of relevant features in the supervectors. All results are based on NIST'02.

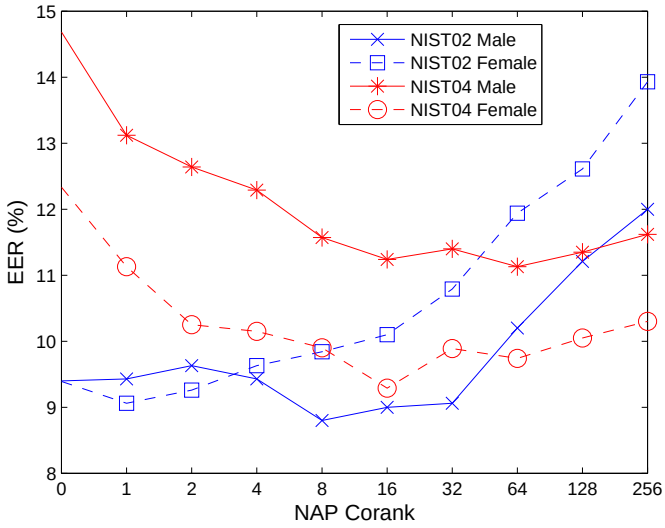


Fig. 4. Effect of varying the NAP corank (number of nuisance dimensions) on the speaker verification performance in NIST'02 and NIST'04 under the core test condition. Corank = 0 means that no NAP was applied.

SVM: A comparative study, Decision Support Systems 48 (1) (2009) 191–201.

[11] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of

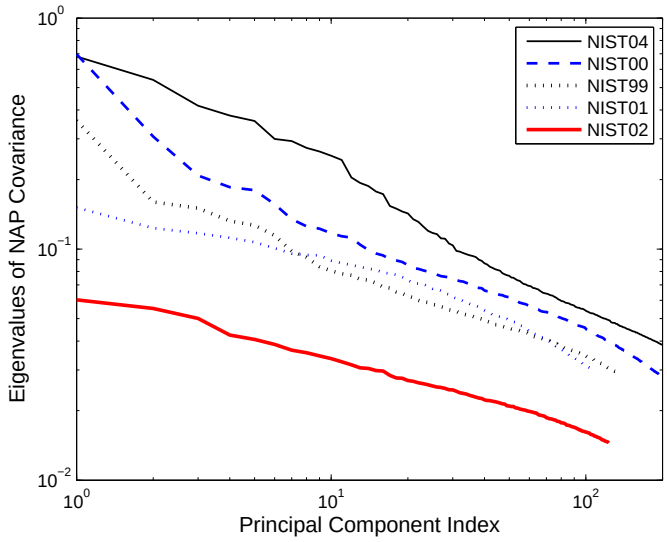
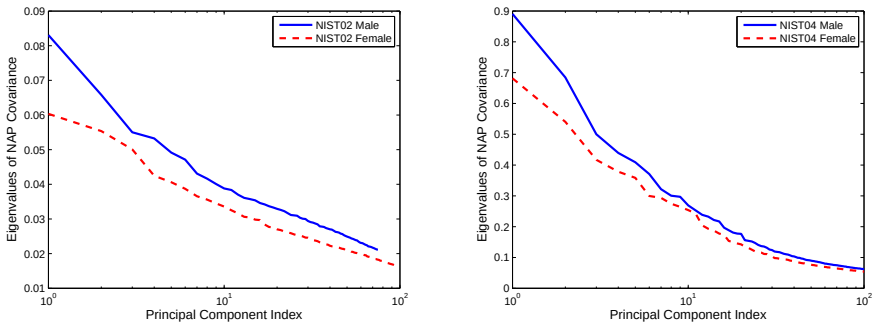


Fig. 5. The eigenvalues of NAP covariance matrices of different corpora. The larger the eigenvalue, the greater the session variation.



(a) (b)

Fig. 6. The eigenvalues of NAP covariance matrices of male and female speakers in (a) NIST'02 and (b) NIST'04. The larger the eigenvalue, the greater the session variation.

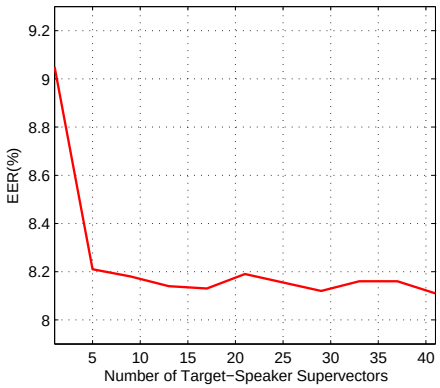
support vector machines, in: Proc. Int. Joint Conf. Artificial Intelligence, 1999, pp. 55–60.

[12] Y. Lin, Y. Y. Lee, G. Wahba, Support vector machines for classification in nonstandard situations, Machine Learning 46 (1-3) (2002) 191–202.

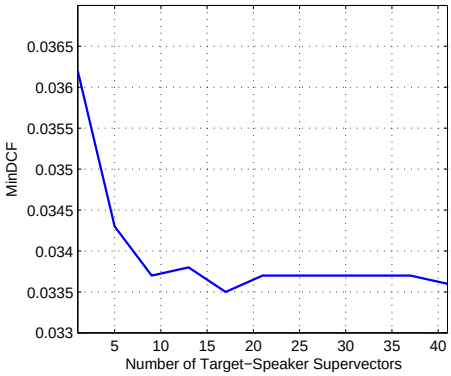
[13] S. B. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. on ASSP 28 (4) (1980) 357–366.

[14] Y. LeCun, F. J. Huang, Loss functions for discriminative training of energy-based models, in: Proc. of the 10-th International Workshop on Artificial Intelligence and Statistics (AIStats'05), 2005.

[15] B. Efron, G. Gong, A leisurely look at bootstrap, the jackknife, and cross-validation, The American Statistician 37 (1) (1983) 36–48.



(a)



(b)

No. of Target-Speaker GSVs	5	9	13	17	21	29	33	36	41
1	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
5	-	0.292	0.098	0.129	0.916	0.148	0.320	0.110	0.532
9	-	-	0.650	0.682	0.185	0.719	0.960	0.602	0.648
13	-	-	-	0.958	0.063	1.000	0.678	0.876	0.396
17	-	-	-	-	**0.002**	1.000	0.590	0.957	0.334
21	-	-	-	-	-	0.068	0.199	0.045	0.394
29	-	-	-	-	-	-	0.628	0.916	0.378
33	-	-	-	-	-	-	-	0.510	0.712
36	-	-	-	-	-	-	-	-	0.233

(c)

Fig. 7. (a) EER and (b) minimum DCF versus number of speaker-class supervectors used for training the speaker-dependent SVMs in NIST'02. The supervectors were obtained by utterance partitioning with acoustic vector resampling (UP-AVR, $N = 4$). (c) p-values of McNemar's tests [26] on the pairwise differences between the EERs in (a). For each entry, $p < 0.005$ means that the difference between the EERs is statistically significant at a confidence level of 99.5%.

[16] B. Fauve, N. Evans, J. Mason, Improving the performance of text-independent short duration SVM- and GMM-based speaker verification, in: Odyssey 2008, 2008.

[17] C. Cieri, D. Miller, K. Walker, The Fisher corpus: A resource for the next generations of speech-to-text, in: Proc. 4th Int. Conf. Lang. Resources Evaluation, 2004, pp. 69–71.

[18] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, Score normalization for text-independent speaker verification systems, Digital Signal Processing 10 (2000) 42–54.

[19] W. M. Campbell, D. E. Sturim, D. A. Reynolds, A. Solomonoff, SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, in: Proc. ICASSP'06, Vol. 1, 2006, pp. 97–100.

[20] A. Solomonoff, W. M. Campbell, I. Boardman, Advances in channel compensation for SVM speaker recognition, in: Proc. of ICASSP'05, 2005, pp.

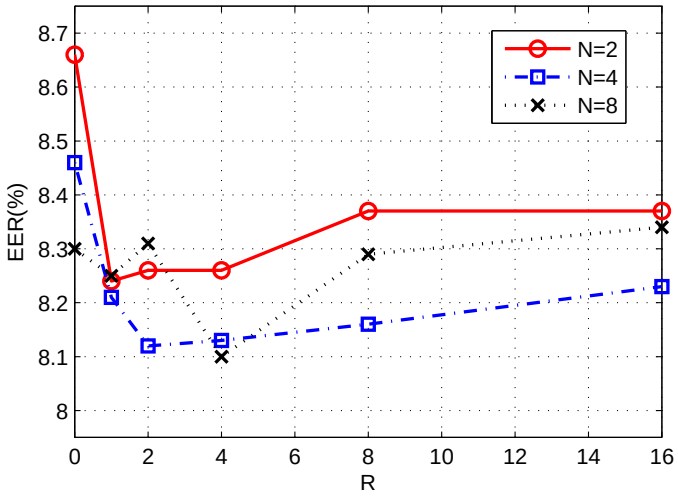


Fig. 8. Performance of UP-AVR for different numbers of partitions (N) and resampling (R) in NIST'02. When $R = 0$, UP-AVR is reduced to UP.

629–632.

[21] B. S. Atal, Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. 55 (6) (1974) 1304–1312.

[22] J. Pelecanos, S. Sridharan, Feature warping for robust speaker verification, in: Proc. Speaker Odyssey, 2001, pp. 213–218.

[23] SVMlight, http://svmlight.joachims.org/.

[24] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, The DET curve in assessment of detection task performance, in: Proc. Eurospeech'97, 1997, pp. 1895–1898.

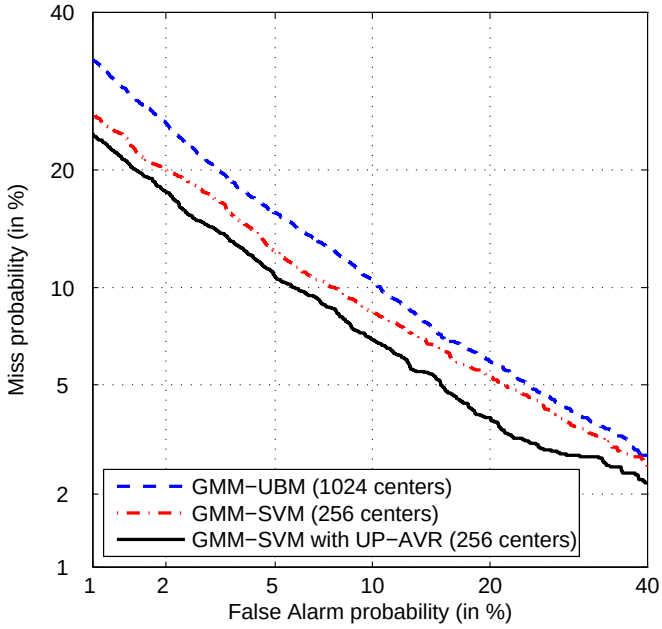
[25] A. Solomonoff, C. Quillen, W. M. Campbell, Channel compensation for SVM speaker recognition, in: Proc. Odyssey: The Speaker and Language Recognition Workshop, Toledo, Spain, 2004, pp. 41–44.

[26] L. Gillick, S. Cox, Some statistical issues in the comparison of speech recognition algorithms, in: Proc. ICASSP'89, 1989, pp. 532–535.

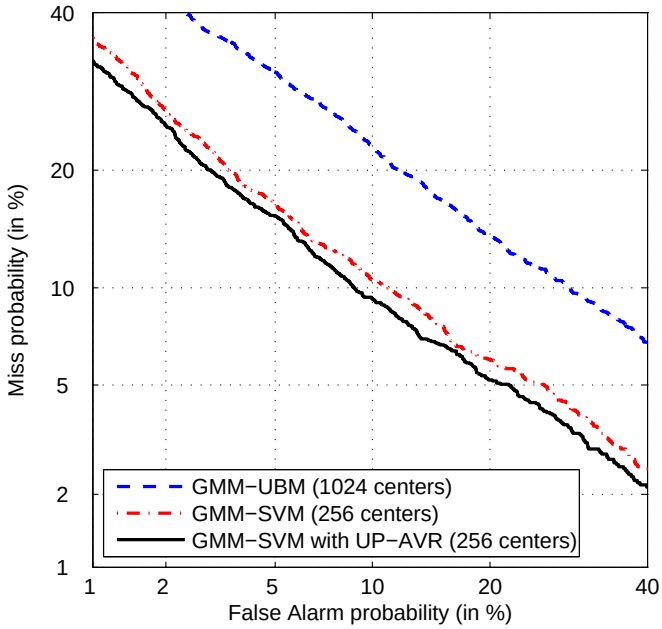
[27] C. Longworth, M. J. F. Gales, Combining derivative and parametric kernels for speaker verification, IEEE Transactions in Audio,Speech and Language Processing 17 (4) (2009) 2009.

[28] G. N. Ramaswamy, A. Navratil, U. V. Chaudhari, R. D. Zilca, The IBM system for the NIST-2002 cellular speaker verification evaluation, in: ICASSP, Vol. 2, 2003, pp. 61–64.

[29] D. A. van Leeuwen, Speaker adaptation in the NIST speaker recognition evaluation 2004, in: Interspeech, 2005, pp. 1981–1984.



(a)

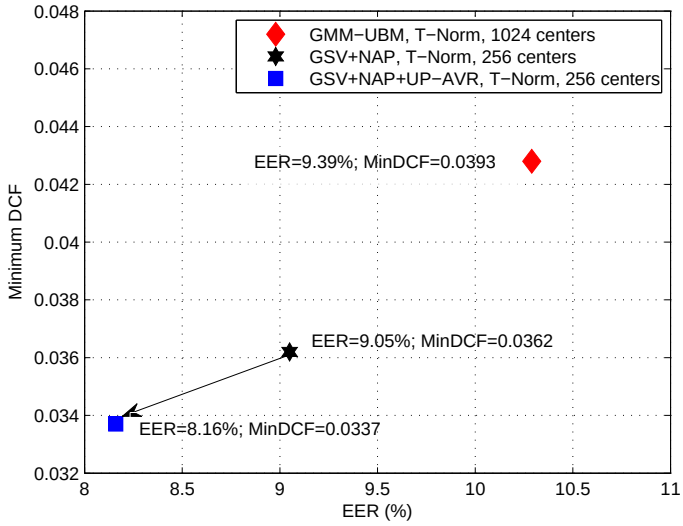


(b)

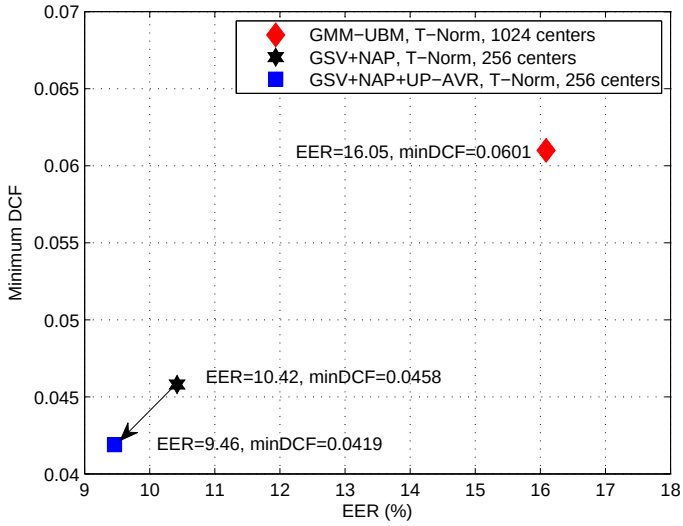
Fig. 9. The DET performance of GMM–UBM, GMM–SVM, and GMM–SVM with utterance partitioning in (a) NIST'02 and (b) NIST'04. T-Norm was applied in all cases.

[30] J. F. Bonastre, F. Wils, S. Meignier, ALIZE, a free toolkit for speaker recognition, in: ICASSP, Vol. 1, 2005, pp. 737–740.

[31] L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, J. Zhen, SRI's



(a) NIST'02



(b) NIST'04

Fig. 10. Minimum DCF versus EER demonstrating the performance improvement obtained by the utterance partitioning approach in (a) NIST'02 and (b) NIST'04.

2004 NIST speaker recognition evaluation system, in: ICASSP, 2005, pp. 173–176.

[32] J. Gonzalez-Rodrigueza, A. Drygajlob, D. Ramos-Castroa, M. Garcia-Gomarc, J. Ortega-Garciaa, Robust estimation, interpretation and assessment of likelihood ratios in forensic speaker recognition, Computer Speech and Language 20 (2-3) (2006) 331–355.

[33] Y. Bar-Yosef, Y. Bistritz, Adaptive individual background model for speaker verification, in: Interspeech, 2009, pp. 1271–1274.

[34] P. Kenny, M. Mihoubi, P. Dumouchel, New MAP estimators for speaker

recognition, in: Eurospeech, Geneva, 2003, pp. 2961–2964.

[35] 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: ICASSP, 2009, pp. 4237–4240.

[36] R. M. Bolle, N. K. Ratha, S. Pankanti, Evaluating authentication systems using bootstrap confidence intervals, in: Proc. AutoID'99, 1999, pp. 9–13.

[37] R. M. Bolle, Guide to biometrics, Springer, New York, 2004.