Boosting the Performance of I-Vector Based Speaker Verification via Utterance Partitioning

Wei RAO and Man-Wai MAK, *Member, IEEE*

***Abstract*—The success of the recent i-vector approach to speaker verification relies on the capability of i-vectors to capture speaker characteristics and the subsequent channel compensation methods to suppress channel variability. Typically, given an utterance, an i-vector is determined from the utterance regardless of its length. This paper investigates how the utterance length affects the discriminative power of i-vectors and demonstrates that the discriminative power of i-vectors reaches a plateau quickly when the utterance length increases. This observation suggests that it is possible to make the best use of a long conversation by partitioning it into a number of sub-utterances so that more i-vectors can be produced for each conversation. To increase the number of sub-utterances without scarifying the representation power of the corresponding i-vectors, repeated applications of frame-index randomization and utterance partitioning are applied. Results on NIST 2010 speaker recognition evaluation (SRE) suggest that (1) using more i-vectors per conversation can help to find more robust linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) transformation matrices, especially when the number of conversations per training speaker is limited; and (2) increasing the number of i-vectors per target speaker helps the i-vector based support vector machines (SVM) to find better decision boundaries, thus making SVM scoring outperforms cosine distance scoring by 19% and 9% in terms of minimum normalized DCF and EER.**

***Index Terms*—Speaker verification, i-vectors, utterance partitioning with acoustic vector resampling (UP-AVR), linear discriminant analysis, support vector machines.**

I. Introduction

In recent years, using i-vectors [1] as features has become one of the promising approaches to text-independent speaker verification. Unlike joint factor analysis (JFA) [2] in which two distinct space (speaker space and channel space) are defined, the i-vector approach defines a single space called total variability space. The acoustic characteristics (including both speaker and channel) of an utterance are represented by a single vector called the i-vector whose elements are essentially the latent variables of a factor analyzer. Compared with the GMM-supervectors, the dimensionality of i-vectors is much lower. Therefore, statistical techniques such as linear discriminant analysis (LDA) [3], within-class covariance normalization (WCCN) [4], and probabilistic LDA [5] can be applied to suppress the channel- and session-variability.

While these techniques have achieved state-of-the-art performance in recent NIST Speaker Recognition Evaluations (SRE), they require multiple training speakers each providing sufficient numbers of sessions to train the transformation matrices or loading matrix. However, collecting such a corpus is expensive and inconvenient. In a typical training dataset, the number of speakers could be fairly large, but the number of speakers who can provide many sessions is quite limited. When the number of training speakers and/or number of recording sessions per speaker are insufficient, numerical difficulty or error will occur in estimating the transformation matrices, resulting in inferior performance. In machine learning literature, this is known as the small sample-size problem [6], [7].

Before i-vectors can be extracted from utterances, it is necessary to use the utterances of a large number of speakers to compute the total variability matrix (the factor loading matrix in factor analysis). Then, given the utterance of a target speaker or a claimed speaker, the latent variables that constitute the i-vector are estimated based on the total variability matrix and the sufficient statistics of the utterance. Therefore, the speaker-dependent information of the whole utterance is embedded in this low-dimensional i-vector. The amount of speaker information will certainly increase with the utterance length but the increase is unlikely to be linear. To confirm this conjecture, we have investigated the relationship between the length of the utterances and the discriminative power of the resulting i-vectors [8]. Interestingly, we observed that the discriminative power of the i-vectors becomes saturated quickly and flatten out when the utterances exceed 2–3 minutes in length.

Intuitively, if the discriminative power of i-vectors becomes saturated when the utterance length reaches two minutes, it will be a waste of resources if conversations longer than two minutes are used for estimating the i-vectors. A better way to exploit the speaker information from a long conversation is to divide it into a number of short conversations (sub-utterances) so that multiple i-vectors can be produced for each conversation. In [9], [10], we developed a partitioning technique, namely utterance partitioning with acoustic vector resampling (UP-AVR), to alleviating the data imbalance problem in GMM-SVM speaker verification. The idea is to partition an enrollment utterance into a number of sub-utterances and to resample the acoustic vectors to produce more GMM-supervectors for training the target-speaker's SVM. Specifically, the frame indexes of a long conversation are firstly randomized; then the randomized frame-sequence is partitioned into equal-length segments, with each segment independently used for estimating a GMM-supervector. This frame-index randomization and partitioning process can be repeated several times to produce a desirable number of GMM-supervectors for each conversation. In this paper, we extend this partitioning and resampling method to i-vector systems.

The authors are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: ellen.wei-rao@polyu.edu.hk; enmwmak@polyu.edu.hk).

The extensions have four fronts:

1) applying UP-AVR to produce more i-vectors per speaker for training the LDA and WCCN matrices and for training target-speaker's SVMs.
2) analyzing the relationship between the length of sub-utterances and the discriminative power of the corresponding i-vectors;
3) investigating the effect of the number of training utterances and training speakers on the discriminative power of LDA and WCCN projected i-vectors;
4) comparing the effectiveness of UP-AVR against the classical pseudo-inverse LDA and PCA pre-preprocessing in alleviating the small-sample size problem that LDA and WCCN may encounter.

In [8], we have reported some preliminary results on Item 1. In this paper, we provide further results on Item 1 and perform extensive analysis on UP-AVR in i-vector systems and the relationship between utterance length and discriminative power of i-vectors. Moreover, unlike our earlier work in [9], [10], in this paper, we address not only the data-imbalance problem in i-vector based SVM scoring but also the small-sample size problem in LDA and WCCN.

As compared to GMM–SVM, there are some new challenges when applying UP-AVR to i-vector systems. First, because UP-AVR produces multiple GMM-supervectors from one enrollment utterance, some of the speaker-class supervectors may be highly similar. If these supervectors are linearly separable from the background-speakers' supervectors,[1] the SVM training algorithm will select those that are closest to the decision boundary as support vectors and consider the rest as redundant vectors. Therefore the existence of highly similar or redundant GMM-supervectors will not be detrimental to the resulting SVMs. However, in the i-vector case, because UP-AVR is used to generate more i-vectors for estimating the LDA and WCCN matrices, all training i-vectors, including redundant and highly similar ones, have contribution to the computation of projection matrices. Therefore, it is important to investigate how these vectors might affect the performance of the i-vector systems. Second, in GMM-SVM, if a sub-utterance is too short (say less than 15 seconds), the resulting GMM-supervector will be almost identical to that of the UBM. The i-vectors, on the other hand, do not enjoy this nice property because of the matrix inversion in Eq. 6 of [1]. Therefore, it is important to investigate the effect of varying the sub-utterance length in the case of i-vector systems.

The paper is organized as follows. Section II outlines the i-vector framework for speaker verification. Section III highlights the relationship between the utterance length and the discriminative power of LDA+WCCN projected i-vectors. Section IV describes the idea of UP-AVR and its applications to the i-vector framework. In Sections V and VI, we report evaluations based on NIST 2010 SRE [11]. Section VII concludes the findings.

[1]This is likely because the dimension of supervectors is much larger than the number of training supervectors.

II. The I-vector Framework For Speaker Verification

The i-vector approach to speaker verification can be divided into three stages: i-vector extraction, intersession compensation and scoring.

A. I-vector Extraction

The i-vector approach is based on the idea of joint factor analysis (JFA) [12]. In [1], Dehak et al. notice that the channel factors in JFA also contain speaker-dependent information. This finding motivates them to model the total variability space (including channels and speakers) instead of modeling the channel- and speaker-spaces separately. Specifically, given an utterance, the speaker- and channel-dependent GMM-supervector [13] $\boldsymbol{m}_s$ is written as:

$$\boldsymbol{m}_s = \boldsymbol{m} + \boldsymbol{T}\mathbf{w}_s \tag{1}$$

where $\boldsymbol{m}$ is the GMM-supervector formed by stacking the mean vectors of the universal background model (UBM) [14] which is speaker- and channel- independent, $\boldsymbol{T}$ is a low-rank total variability matrix, and $\mathbf{w}_s$ is a low-dimensional vector called the i-vector. The training of the total variability matrix is almost identical to that of the eigenvoice matrix in JFA. The only difference is that the utterances of a training speaker are considered to be produced by different speakers.

B. Inter-session Compensation

Because i-vectors contain both speaker and channel variation in the total variability space, inter-session compensation plays an important role in the i-vector framework. It was found in [1] that projecting the i-vectors by linear discriminant analysis followed by within class covariance normalization achieves the best performance.

1) Linear Discriminant Analysis: LDA is commonly used for dimensionality reduction. The idea is to find a set of orthogonal axes for minimizing the within-class variation and maximizing the between-class separation. In the i-vector framework, the i-vectors of a speaker constitute a class, leading to the following objective function for multi-class LDA [3]:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmax}} \left\{ \operatorname{tr} \left[\left(\mathbf{A}^{\mathsf{T}} \mathbf{S}_w \mathbf{A} \right)^{-1} \left(\mathbf{A}^{\mathsf{T}} \mathbf{S}_b \mathbf{A} \right) \right] \right\} \tag{2}$$

where $\hat{\mathbf{A}}$ comprises the optimal subspace to which the i-vectors should be projected, $\mathbf{S}_w$ is the within-speaker scatter matrix, and $\mathbf{S}_b$ is the between-class scatter matrix. These two scatter matrices are written as follows:

$$\mathbf{S}_w = \sum_{i=1}^{S} \frac{1}{M_i} \sum_{j=1}^{M_i} (\mathbf{w}_j^i - \boldsymbol{\mu}^i)(\mathbf{w}_j^i - \boldsymbol{\mu}^i)^{\mathsf{T}} \tag{3}$$

and

$$\mathbf{S}_b = \sum_{i=1}^{S} (\boldsymbol{\mu}^i - \boldsymbol{\mu})(\boldsymbol{\mu}^i - \boldsymbol{\mu})^{\mathsf{T}} \tag{4}$$

where

$$\boldsymbol{\mu}^i = \frac{1}{M_i} \sum_{j=1}^{M_i} \mathbf{w}_j^i \tag{5}$$

is the mean i-vector of the i-th speaker, S is the number of training speakers, M_i is the number of utterances from the i-th training speaker, and $\boldsymbol{\mu}$ is the global mean of all i-vectors in the training dataset. Eq. 2 leads to the projection matrix $\hat{\mathbf{A}}$ that comprises the leading eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$.

2) Within Class Covariance Normalization: Within Class Covariance Normalization (WCCN) [4] was originally used for normalizing the kernels in SVMs. In the i-vector framework, WCCN is to normalize the within-speaker variation. Dehak et al. [1] found that the best approach is to project the LDA reduced i-vectors to a subspace specified by the square-root of the inverse of the following within-class covariance matrix:

$$\mathbf{W} = \sum_{i=1}^{S} \frac{1}{M_i} \sum_{j=1}^{M_i} (\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}_j^i - \widetilde{\boldsymbol{\mu}^i})(\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}_j^i - \widetilde{\boldsymbol{\mu}^i})^{\mathsf{T}} \tag{6}$$

where

$$\widetilde{\boldsymbol{\mu}^i} = \frac{1}{M_i} \sum_{j=1}^{M_i} \hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}_j^i \tag{7}$$

and $\hat{\mathbf{A}}$ is the LDA projection matrix found in Eq. 2. The WCCN projection matrix $\mathbf{B}$ can be obtained by Cholesky decomposition of $\mathbf{W}^{-1} = \mathbf{B}\mathbf{B}^{\mathsf{T}}$.

C. Scoring Methods

1) Cosine Distance Scoring: Cosine distance scoring (CDS) [15] is commonly used in the i-vector framework. This scoring approach is computationally efficient. The method computes the cosine distance score between the claimant's i-vector ($\mathbf{w}^{(c)}$) and target-speaker's i-vector ($\mathbf{w}^{(s)}$) in the LDA+WCCN projection space:

$$S_{\cos}\left(\mathbf{w}^{(c)}, \mathbf{w}^{(s)}\right) = \frac{\left\langle \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(c)}, \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(s)} \right\rangle}{\left\| \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(c)} \right\| \left\| \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(s)} \right\|} \tag{8}$$

The score is then further normalized (typically by ZT-norm [16]) before comparing with a threshold for making a decision.

2) Support Vector Machine Scoring: The idea of support vector Machine (SVM) scoring [15] is to harness the discriminative information embedded in the training data by constructing an SVM that optimally separates the i-vectors of a target speaker from those of background speakers. Unlike cosine distance scoring, the advantage of SVM scoring is that 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. Given the SVM of target speaker s, the verification score of claimant c is given by

$$S_{\text{SVM}}(\mathbf{w}^{(c)}, \mathbf{w}^{(s)}) = \alpha_0^{(s)} K\left(\mathbf{w}^{(c)}, \mathbf{w}^{(s)}\right) - \sum_{i \in \mathcal{S}^{(b)}} \alpha_i^{(s)} K\left(\mathbf{w}^{(c)}, \mathbf{w}^{(b_i)}\right) + d^{(s)} \tag{9}$$

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 $\mathbf{w}^{(b_i)}$ is the i-vectors 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. It was found [1] that the cosine kernel is appropriate. Specifically,

$$K\left(\mathbf{w}^{(c)}, \mathbf{w}^{(s)}\right) = \frac{\left\langle \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(c)}, \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(s)} \right\rangle}{\left\| \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(c)} \right\| \left\| \mathbf{B}^{\mathsf{T}}\hat{\mathbf{A}}^{\mathsf{T}}\mathbf{w}^{(s)} \right\|} \tag{10}$$

where we replace $\mathbf{w}^{(s)}$ by $\mathbf{w}^{(b_i)}$ for evaluating the second term of Eq. 9. Note that Eq. 8 and Eq. 10 are the same. However, their role in the scoring process is different. The former is directly used for calculating the score, whereas the latter is used for kernel evaluation.

While SVM scoring can take the background speakers' i-vectors into consideration, its major shortcoming is that the SVM decision boundary is mainly governed by the background speakers' i-vectors because there is only one target-speaker's i-vector to define the decision boundary. This situation is known as training data-imbalance [17], [18]. We have recently proposed a method called utterance partitioning to alleviate this problem, which will be described in details in Section IV.

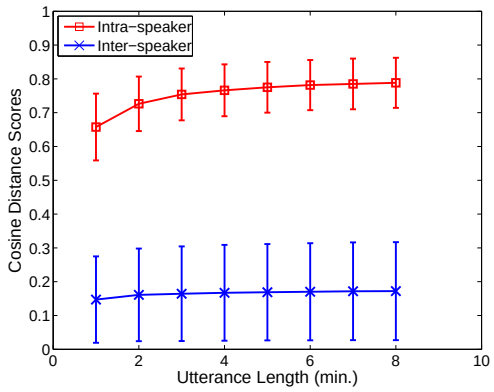
III. Effect of Utterance Length on I-Vectors

The major advantage of the i-vector framework is that a variable-length utterance can now be represented by a low-dimensional i-vector. This low-dimensional space facilitates the application of LDA and WCCN, which require low-dimensionality to ensure numerical stability (unless abundant training data are available). As the i-vectors are very compact, it is interesting to investigate if short utterances are still able to maintain the discriminative power of i-vectors. To this end, we computed the intra- and inter-speaker cosine-distance scores of 272 speakers extracted from the interview_mic and phonecall_tel sessions of NIST 2010 SRE.

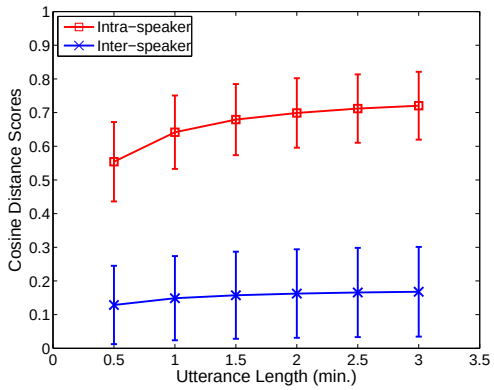
Each conversation of these speakers was divided into a number of equal-length segments. Then, sub-utterances of variable length were obtained by concatenating variable numbers of equal-length segments. A voice activity detector (VAD) [19] was applied to extract the acoustic vectors corresponding to the speech regions in the sub-utterances. The acoustic vectors of each sub-utterance were then used for estimating an i-vector, followed by LDA and WCCN projections to 150-dim i-vectors, which were used for computing the cosine distance scores.

Fig. 1 shows the mean intra- and inter-speaker scores (with error bars indicating two standard deviation) of the three types of speech. For "8-min interview_mic", the scores were obtained from the 8-min interview sessions of 29 male speakers in NIST 2010 SRE, each providing 4 interview conversations. This amounts to 174 intra-speaker scores and 6,496 inter-speaker scores for each utterance length. For "3-min interview_mic", the scores were obtained from the 3-min interview sessions of 196 male speakers, each providing 4 interview conversations. This amounts to 1,176 intra-speaker scores and 305,760 inter-speaker scores for each utterance length. For "5-min phonecall_tel", the scores were obtained

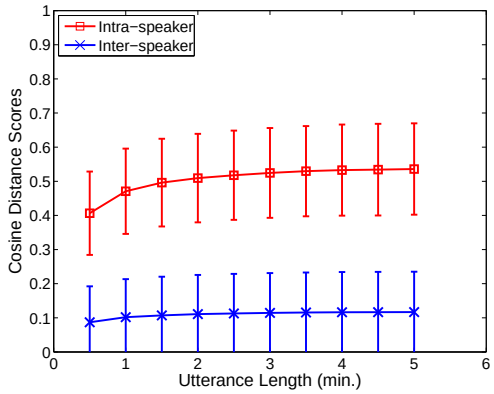
[2]We assume one enrollment utterance per target speaker.



(a) 8-min interview_mic



(b) 3-min interview_mic



(c) 5-min phonecall_tel

Fig. 1. Intra-speaker and inter-speaker cosine-distance scores versus utterance length for (a) 8-min interview conversation, (b) 3-min interview conversation, and (c) 5-min telephone conversation.

from the 5-min phonecall conversations of 47 male speakers, each providing 4 conversations. This amounts to 282 intra-speaker scores and 17,296 inter-speaker scores for each utterance length. Evidently, both types of scores flatten out after the segment length used for estimating the i-vectors exceeds a certain threshold.

To further analyze the discriminative power of i-vectors with respect to the utterance length, we plot in Fig. 2 the minimum decision cost (MinDCF) versus the utterance length for estimating the i-vectors using the intra- and inter-speaker scores shown in Fig. 1. The lower the cost, the higher the discriminative power of the i-vectors. The result clearly

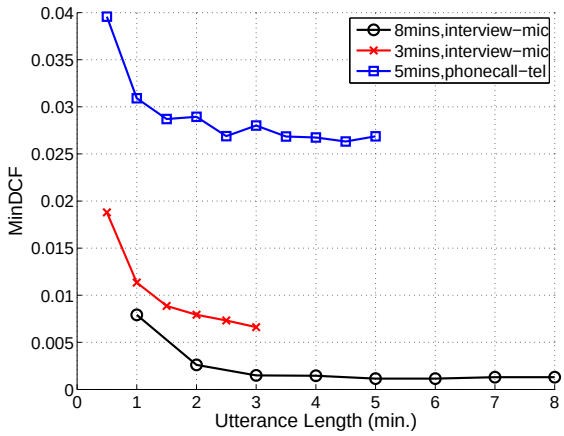


Fig. 2. Minimum decision costs (old MinDCF) achieved by i-vectors derived from utterances of various lengths. The costs were based on the intra- and inter-speaker cosine-distances shown in Fig. 1. For computation efficiency, no score normalization was applied.

suggests that the discriminative power becomes saturated for utterance length exceeding 2 minutes. This finding suggests that it is not necessary to record very long utterances for the i-vectors to achieve good performance. From another perspective, if long conversations are already available, it may be beneficial to divide the long conversations into a number of sub-utterances to produce more i-vectors per conversation. This can be achieved by our recently proposed utterance partitioning method to be described next.

IV. Utterance Partitioning with Acoustic Vector Resampling

Utterance partitioning with acoustic vector resampling (UP-AVR) [10] was proposed to maximize the utilization of target-speaker's information and to increase the influence of speaker-class data on the SVM decision boundary. In the current work, UP-AVR is applied to partition a conversation into a number of sub-utterances, each producing one i-vector.

A. Procedure of UP-AVR

To produce a sufficient number of sub-utterances without compromising their representation power, UP-AVR uses the notion of random resampling in bootstrapping [20]. The idea is based on the fact that changing the order of acoustic vectors will not affect the resulting i-vector. Fig. 3 illustrates the procedure of UP-AVR. For each conversation, a sequence of acoustic vectors (see Section V-A) is extracted. Then, the sequence is partitioned into N equal-length segments, and an i-vector is estimated from each segment. Obviously, the number of i-vectors increases when the segment length decreases. However, decreasing the segment length will inevitably compromise the representation power of the resulting i-vectors. To increase the number of i-vectors for each utterance but maintaining their representation power, the order of the acoustic vectors in the sequence is randomly rearranged and the partitioning process is repeated. If this partitioning-randomization process is repeated R times, $RN + 1$ i-vectors

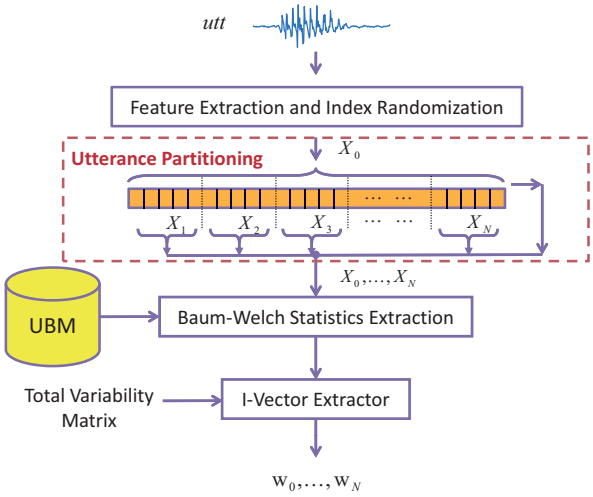


Fig. 3. The procedure of utterance partitioning with acoustic vector resampling (UP-AVR). Note that index randomization, utterance partitioning, and i-vector extraction can be repeated several times to obtain a sufficient number of i-vectors.

can be obtained from a single conversation, where the additional one is obtained from the entire acoustic sequence. In theory, we can obtain an infinite number of i-vectors when $R \rightarrow \infty$. However, when R increases, the segments will contain many acoustic vectors that are identical to each other, resulting in many similar i-vectors. Similar situation occurs if two or more segments are used for estimating an i-vector. To avoid this situation, R should be small. In this work, R was limited to 4. Section VI-C analyzes the effect of varying R and N on the discriminative power of i-vectors.

B. UP-AVR for LDA and WCCN

LDA requires a sufficient number of recording sessions per training speaker for estimating the within-speaker and between-speaker scatter matrices. However, collecting such recordings is costly and inconvenient. As demonstrated in Section III, when the utterance length is sufficiently long, further increasing the length will not increase the i-vectors' discriminative power significantly. Therefore, given a long conversation, some intrinsic speaker information will be wasted if the whole conversation is used for estimating the i-vector. To make a better use of the long conversation, we can apply UP-AVR to produce more i-vectors for estimating the LDA and WCCN projection matrices. It helps the LDA to find a subspace with less intra-speaker variation by alleviating the numerical problem.

C. UP-AVR for SVM Scoring

A simple strategy for solving the data imbalance problem in SVM scoring is to increase the number of minority-class samples for training the SVMs [18]. One may use more enrollment utterances, which means more i-vectors from the speaker class. However, this strategy shifts the burden to the client speakers by requesting them to provide multiple enrollment utterances, which may not be practical. With UP-AVR, a number of i-vectors can be produced for training the target-speaker dependent SVM even if the target-speaker provides only one enrollment utterance, which can enhance the influence of the target-speaker data on the SVM's decision boundary.

An alternative approach to alleviating the data-imbalance problem is to train a speaker-independent SVM to classify a pair of utterances as belonging to the "same speaker" or to "different speakers" [21], [22]. Because many same-speaker pairs can be obtained from training data, data-imbalance will not be an issue. The downside is that the method requires a lot more computation resources for training the speaker-independent SVM.

D. Properties of Generated I-Vectors

It is of interest to investigate the statistical properties of the generated i-vectors when the values of N and R vary. To this end, we selected one hundred 3-min interview utterances from NIST 2010 SREs and estimate the i-vectors produced by UP-AVR for different N and R, followed by LDA and WCCN projection to 150-dimensional vectors. Then, for each full-length utterance, we computed the cosine distance scores between the i-vector derived from the full-length utterance and the RN i-vectors generated by UP-AVR. We also computed the cosine distance scores among these generated i-vectors. The scores across all of the 100 full-length utterances are then averages, which results in two sets of average scores: one representing the similarity between full-length utterances and sub-utterances and another representing the similarity among sub-utterances.

Fig. 4 shows how these scores vary with respect to the number of partitions (N) per conversation while keeping R fixed. The figure clearly suggests that when $N = 2$ (i.e., sub-utterances are long), the i-vectors of sub-utterances are similar to that of the full-length utterances. They are also similar among themselves. The similarity decreases but the score variances increase when the number of partitions increases, i.e., sub-utterances become shorter. This is reasonable because when N is small, the acoustic vectors of a sub-utterance are identical to a large portion of the acoustic vectors in the full-length utterance. When N increases, the sub-utterances become shorter, resulting in lower similarity but higher variability with respect to each others.

Fig. 5 shows the effect of varying the number of resampling R on these two sets of scores when N is fixed. As opposed to Fig. 4, when R increases, the similarity among the sub-utterances of a full-length utterance increases but their score variances decrease. The reason is that after several cycles of resampling, the sub-utterances in the current resampling cycle will contain some acoustic vectors that have already appeared in the sub-utterances of the previous resampling cycles, causing higher similarity among the i-vectors of the sub-utterances. However, these sub-utterances are still not identical and they play important role in the training of target-speaker SVMs, which will be further elaborated in Section VI-D.

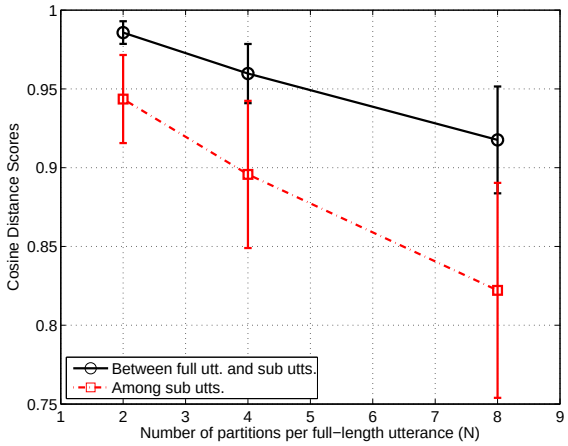


Fig. 4. Average cosine distance scores with error bars versus the number of partitions per full-length utterance. In all cases, the number of re-sampling in UP-AVR was set to 1, i.e. $R = 1$.

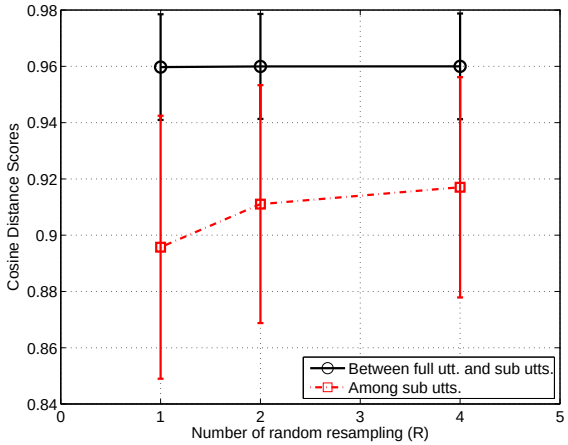


Fig. 5. Average cosine distance scores with error bars versus the number of random resampling R. In all cases, the number of partitions in UP-AVR was set to 4, i.e. $N = 4$.

V. Experimental Setup

A. Speech Data and Acoustic Features

The *extended core set* of NIST 2010 Speaker Recognition Evaluation (SRE) was used for performance evaluation. This paper focuses on the interview and microphone speech of the extended core task, i.e., Common Conditions 1, 2, 4, 7 and 9. Table I shows the number of trials and speech types for training and testing under these common conditions. In CC1, the same microphone was used for recording both training and test segments, whereas in CC2 different microphones were used. In CC7, high vocal-effort speech segments were used for testing, whereas in CC9 low vocal-effort segments were used. The equal error rate (EER) and the new minimum Detection Cost Function (DCF) were used as performance indicators.

NIST 2005–2008 SREs were used as development data (UBM, total variability subspace training, LDA, WCCN, T-norm, and ZT-norm). Only the interview and microphone speech of male speakers in these corpora were used. An in-house VAD [19] was applied to detect the speech regions of each utterance. Briefly, for each conversation side, the VAD uses spectral subtraction with a large over-subtraction factor to remove the background noise. The low energy and high energy regions of the noise-removed speech were used for estimating a decision threshold. This energy-based threshold was then applied to the whole utterance to detect the speech regions. Mel-frequency cepstral coefficients (MFCCs) were then extracted from the speech regions of the original noisy signals. Cepstral mean normalization [23] was then applied to the MFCCs, followed by feature warping [24] using a window of 3 seconds. 19 MFCCs together with energy plus their 1st- and 2nd- derivatives were extracted from the speech regions of each utterance, leading to 60-dim acoustic vectors.

B. Total Variability Modeling and Channel Compensation

The i-vector systems are based on a gender-dependent UBM with 1024 mixtures. 9,511 utterances from NIST 2005–2008 SREs were selected for estimating a total variability matrix with 400 total factors. Joint factor analysis (JFA) Matlab code from Brno University of Technology (BUT) [25] was modified for i-vector training and scoring. Before calculating the verification scores, LDA and WCCN projections were performed for channel compensation. We selected 6,102 utterances from 191 speakers in NIST 2005–2008 SREs to estimate the LDA and WCCN matrices. After LDA and WCCN projections, the dimension of i-vectors was reduced to 150.

C. Scoring Methods and Score Normalization

We adopted two scoring methods: cosine distance scoring and SVM scoring. To train the speaker-dependent SVMs, we selected 633 speakers from NIST 2005–2008 SREs as impostor-class data. ZT-norm [16] was used for score normalization. Specifically, 288 T-norm utterances and 288 Z-norm utterances (each from a different set of speakers) were selected from the interview and microphone speech in NIST 2005–2008 SREs.

D. Utterance Length after Utterance partitioning

Fig. 1 and Fig. 2 demonstrate that the discriminative power of i-vector increases most rapidly from 0.5 to 1.0 minute and becomes saturated after 2 minutes. This information gives us some guidelines on how to partition an utterance. Therefore, except for the experiments investigating the effect of the number of partitions, we partitioned all utterances into four segments. This amounts to sub-utterances length of 0.75, 1.25, and 2 minutes for the 3-min, 5-min, and 8-min utterances. While the sub-utterance length for 3-min utterances is relatively short, partitioning the 3-min utterances into 4 segments gives us more i-vectors for training the LDA+WCCN matrices. The results suggest that this is a reasonable compromise between the number of i-vectors and their discriminative power.

VI. Results and Discussions

A. Effects of Training-set Size on I-Vectors

This experiment is to analyze the effect of the number of training utterances and training speakers on the discriminative power of LDA and WCCN projected i-vectors.

TABLE I
THE NUMBER OF MALE TARGET-SPEAKER TRIALS AND IMPOSTOR TRIALS AND THE SPEECH TYPES FOR TRAINING AND TESTING UNDER THE COMMON CONDITIONS THAT INVOLVE MICROPHONE RECORDINGS IN NIST 2010 SRE.

Common Condition		CC1	CC2	CC4	CC7	CC9	Mic
Speech Type	Train	int-mic	int-mic	int-mic	phn-mic	phn-mic	int-mic & phn-mic
	Test	int-mic	int-mic	phn-mic	phn-mic	phn-mic	int-mic & phn-mic
No. of target trials		1,978	6,932	1,886	179	117	11,092
No. of impostor trials		346,857	1,215,586	364,308	39,898	29,667	1,996,316

Mic: Combining the trials of all common conditions that involve microphone recordings. *phn-mic*: Telephone conversation recorded by microphones. *int-mic*: Interview sessions recorded by microphones.

The training set comprises the i-vectors of 191 male speakers in NIST 2005–2008 SRE, with each speaker having 10 i-vectors (sessions). For each experiment, a subset of i-vectors was extracted from this training set to train the LDA and WCCN projection matrices. More precisely, the numbers of i-vectors per speaker were set to 6, 8, and 10. For each configuration, the number of speakers S was progressively increased from 60 (80 when there are only 6 utterances per speaker)[3] to 191. The resulting LDA+WCCN matrices were then used to project two thousand 400-dim i-vectors extracted from 90 speakers in NIST 2010 SRE to i-vectors of dimensions $S-1$ or 150, whichever is less.[4] The discriminative power of the projected i-vectors was quantified by minimum DCF derived from 22,198 intra- and 1.9 million inter-speaker cosine-distance scores without score normalization.

Fig. 6 shows the minimum DCF achieved by the projected i-vectors when the number of speakers and the number of utterances per speaker used for training the LDA+WCCN projection matrices increase. The results suggest that when the number of utterances per speaker is small ($\leq$ 8) the discriminative power of i-vectors *generally* increases when the number of speakers used for training the transformation matrices increases. The increase is more prominent when the number of utterances per speaker is very small (say 6), suggesting that more speakers are required when the number of utterances per speaker is very small. However, when the number of utterances per speaker is sufficiently large (say 10), increasing the number of speakers does not bring significant benefit until the number of speakers is larger than 105. Among the three different numbers of utterances per speaker, using 10 utterances per speaker achieves the lowest minimum DCF regardless of the number of speakers used for training the transformation matrices, suggesting that it is better to use more utterances per speaker than using more speakers but less utterances per speaker. The small fluctuation in minDCF suggests that the channel variability of some speakers in NIST 2005–2008 SREs may not match the channel variability in NIST 2010 SRE, causing slight performance degradation when these speakers were added to the training pool.

[3]When the number of utterances per speaker was limited to 6 and the number of speakers is smaller than 80, the within-class covariance matrix $\mathbf{S}_w$ is close to singular, causing numerical difficulty in estimating the projection matrices.

[4]Because rank $\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right) = \min\{400, S-1\}$.

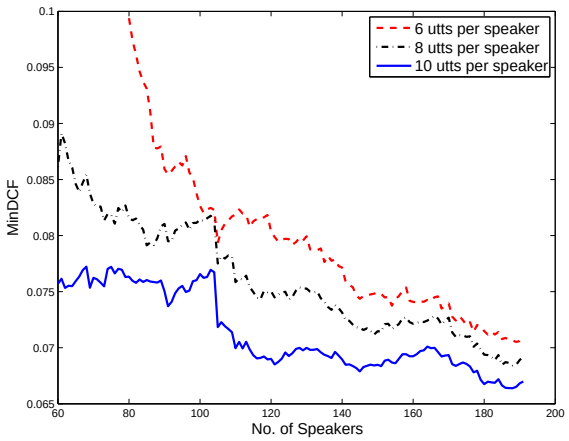


Fig. 6. Minimum decision cost (old MinDCF) versus the number of speakers used for estimating the LDA and WCCN projection matrices. Each speaker has either 6, 8, or 10 utterances for estimating the i-vectors.

B. Small Sample-Size Problem in LDA and WCCN

The numerical difficulty in estimating the LDA and WCCN transformation matrices is due to insufficient rank in the within-speaker scatter matrix (Eq. 3) when the training set size is small. We have investigated two classical approaches to alleviating this small sample-size problem [26]. They are pseudo-inverse LDA and PCA+LDA.

1) *Pseudo-inverse LDA*. The rank deficiency problem can be avoided by replacing the inverse of the within-speaker scatter matrix by its pseudo inverse [27], [28]. The idea is that during singular value decomposition, any components with singular values smaller than a threshold will be automatically discarded by the pseudo-inverse procedure.
2) *PCA+LDA*. We used PCA to project the training i-vectors to a lower dimension space prior to computing the within-speaker scatter matrix [29], [30]. With the reduction in the i-vector dimension, the rank requirement of LDA and WCCN can be reduced to a comfortable level for reliable estimation of the LDA and WCCN transformation matrices.

Table II shows the performance achieved by different approaches to alleviating the small-sample size problem when the number of recording sessions per training speaker (M) increases from 2 to 8 or above. The performance is obtained

by concatenating the scores under Common Conditions 1, 2, 4, 7, and 9 in NIST 2010 SRE. The performance achieved by "Without LDA and WCCN" is considered as the baseline. For "LDA+WCCN", the performance is very poor when $M \leq 3$, because the within-speaker scatter matrix is close to be singular. Only when $M \geq 4$, the benefit of LDA+WCCN becomes apparent. These observations also agree with the findings in [31].

Table II also shows the following properties:

1) when $M \leq 3$, pseudo-inverse LDA can help to avoid the singularity problem. However, this method leads to i-vectors that perform even poorer than those without LDA+WCCN projections. When the within-class scatter matrices have full rank ($M \geq 4$), the performance of pseudo-inverse LDA is the same as the classical LDA.
2) Preprocessing the i-vectors by PCA can not only avoid the singularity problem but also help the LDA to find a better projection matrix. However, when the rank of within-class scatter matrices is too low (e.g., when $M = 2$), the performance of PCA preprocessing is poorer than those without LDA+WCCN projections. Moreover, the effect of PCA diminishes when the number of recordings per training speaker is sufficient ($M \geq 8$).
3) UP-AVR is an effective way to produce more informative i-vectors from a single utterance, thus effectively avoiding the singularity problem in LDA. It also achieves the best performance among all methods investigated.

Fig. 7 shows the effect of varying the dimension of PCA projection on the performance of PCA+LDA. The results suggest that when the number of sessions per speaker (M) is equal to two, PCA cannot help the LDA for all projection dimension. In fact, the performance is even poorer than that without LDA (dotted line). This is caused by insufficient data for training the LDA, even though PCA can help solving the singularity problem. The result also suggest that setting the PCA projection dimension close to the rank of within-class scatter matrices is not a good idea when $M \leq 3$.

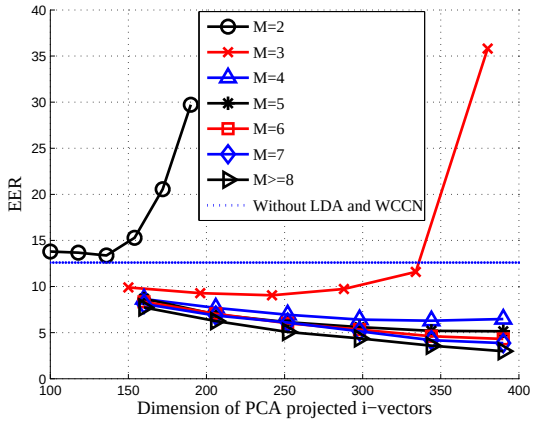


Fig. 7. EER versus the dimension after PCA projection. $M = x$ means each speaker only has x recordings for training the LDA and WCCN matrices.

C. UP-AVR for LDA and WCCN

This experiment investigates the effectiveness of UP-AVR for solving the singularity problem in LDA. Similar to Section VI-B, the number of recording sessions per training speaker was increased from 2 to 8 and above. The results in Table II show that when UP-AVR is applied to increase the number of i-vectors per training speaker, the performance of LDA+WCCN improves significantly. Although many of the i-vectors produced by UP-AVR are extracted from the sub-utterances of the same recording sessions, they possess sufficient speaker-dependent information for training the LDA and WCCN projection matrices and can help LDA to find a subspace with less intra-speaker variation by alleviating the numerical problem. Nevertheless, the contribution of UP-AVR to LDA and WCCN diminishes when the number of recordings per training speaker is sufficient (over 8 per speaker in our experiments).

Figs. 8(a) and 8(b) depict the trend of EER and minimum DCF when the number of recording sessions per speaker and the number of i-vectors per recording session for training the LDA and WCCN matrices increase. The results demonstrate that the most significant performance gain is obtained when the number of i-vectors per recording session increases from 1 to 5, and the performance levels off when more i-vectors are added.

Fig. 9 shows the performances of UP-AVR for different numbers of partitions (N) and resampling (R) for different numbers of recordings per speaker. According to Fig. 9, when the number of recordings per speaker (M) is less than five, increasing the number of partitions and resampling times can improve the performance. However, when $M \geq 6$, the effect of varying N and R diminishes, suggesting that UP-AVR is most effective for training the LDA and WCCN matrixes when the number of recording sessions per speaker is very limited.

D. UP-AVR for SVM Scoring

In this experiment, we used all of the available interview and microphone speech from NIST 2005–2008 SRE to train the LDA and WCCN matrices. The focus of the experiment is on comparing SVM scoring against cosine distance scoring.

Table III compares the performance between SVM scoring and cosine distance scoring. Table III shows that the performance of SVM scoring is slightly worse than that of cosine distance scoring. This may be caused by the data imbalance problem in SVM training. However, after applying UP-AVR to SVM training, the performance of SVM improves. More specifically, increasing the number of target-speakers i-vectors from one i-vector per target-speaker to 9 i-vectors per target-speaker reduces the EER of SVM scoring from 3.26% to 2.71%, which amounts to 17% relative reduction. Similarly, the method reduces the minimum DCF from 0.52 to 0.51, which amounts to 2% relative reduction. This performance improvement makes SVM scoring outperforms cosine distance scoring significantly, as evident by the results (CDS versus SVM+UP-AVR) in Table III. Specifically, when UP-AVR was applied to SVM scoring, the EER and minimum DCF reduce

TABLE II
THE PERFORMANCE OF USING DIFFERENT METHODS FOR SOLVING THE SMALL SAMPLE-SIZE PROBLEM IN LDA AND WCCN.

Methods	EER (%)							MinNDCF						
	M=2	M=3	M=4	M=5	M=6	M=7	$M\geq$8	M=2	M=3	M=4	M=5	M=6	M=7	$M\geq$8
Without LDA and WCCN	12.60							0.90						
LDA + WCCN	23.39	22.25	6.98	5.51	4.59	4.22	2.98	1.00	1.00	0.87	0.81	0.76	0.75	0.63
PI-LDA + WCCN	19.02	20.90	6.98	5.51	4.59	4.22	2.98	0.99	1.00	0.87	0.81	0.76	0.75	0.63
PCA + LDA + WCCN	13.37	9.05	6.29	5.14	4.32	3.86	2.98	1.00	0.95	0.88	0.82	0.77	**0.73**	**0.63**
UP-AVR(2) + LDA + WCCN	6.64	5.78	4.99	4.52	4.08	3.90	2.94	**0.91**	**0.87**	**0.83**	**0.79**	**0.75**	0.74	0.66
UP-AVR(4) + LDA + WCCN	**6.16**	**5.09**	**4.46**	4.05	**3.85**	3.68	**2.90**	0.93	0.89	0.85	0.79	0.76	0.75	0.66
UP-AVR(8) + LDA + WCCN	6.23	5.09	4.48	**3.88**	3.87	**3.65**	2.97	0.92	0.89	0.86	0.80	0.78	0.76	0.69

$M = x$ means each speaker only has x recordings for training the LDA and WCCN matrices. $M \geq 8$ means each speaker provides at least 8 recordings, with an average of 31 recordings per speaker. "LDA": the conventional LDA; "PI-LDA": pseudo-inverse LDA; "PCA + LDA": perform PCA before LDA; "UP-AVR(N)": dividing each of the full-length training utterances into N partitions using UP-AVR, with the number of re-sampling R set to 4.

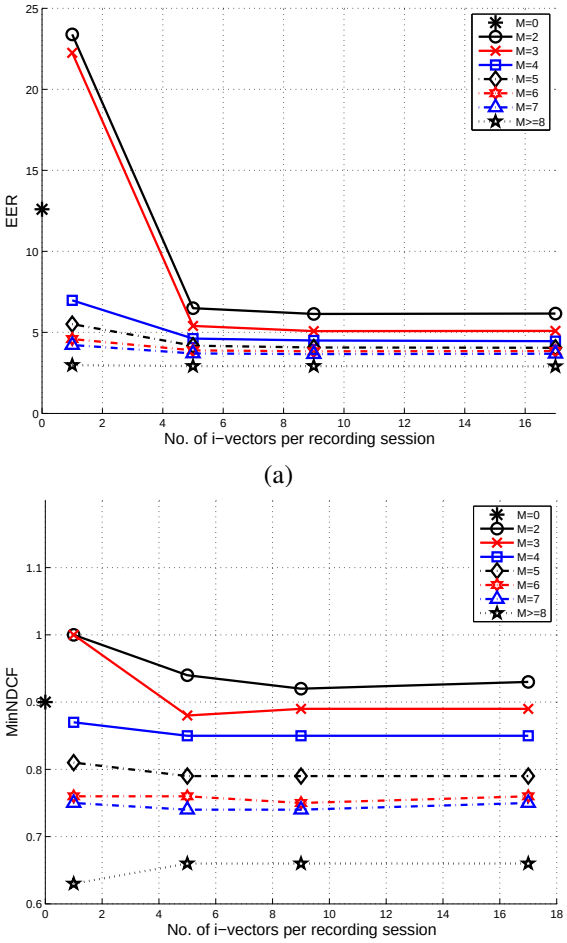


Fig. 8. (a) EER and (b) minimum DCF versus number of i-vectors per recording session for different numbers of recording session per speaker for training the LDA and WCCN matrices. The i-vectors were obtained by utterance partitioning with acoustic vector resampling (UP-AVR, $N = 4$; $R = 1, 2, 4$). M is the number of recordings per speaker used for training the matrices, $M = 0$ means without LDA and WCCN, and $M \geq 8$ means at least 8 utterances per speaker were used for training.

to 2.71% and 0.51, respectively, which amount to 9% and 19% relative reduction.

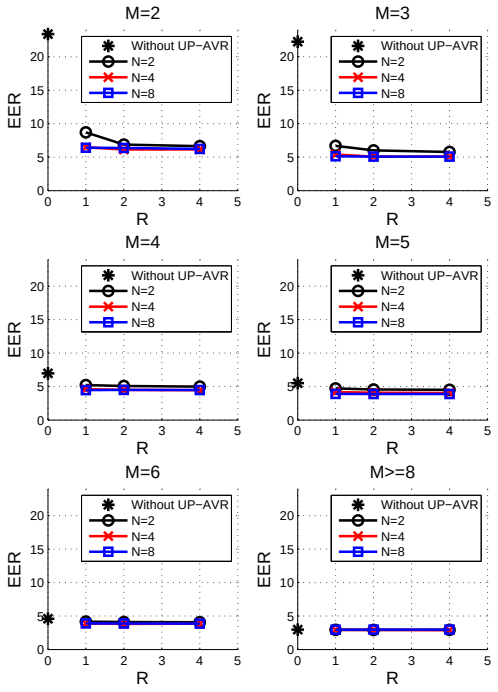


Fig. 9. The effect of varying the number of partitions (N) and the number of resampling (R) on the performance of UP-AVR. $R = 0$ means without applying UP-AVR to the utterances.

Note that UP-AVR can also be applied to cosine distance scoring. Specifically, instead of training an SVM for each target speaker, we used the RN i-vectors produced by UP-AVR together with the one estimated from the full-length enrollment utterance to represent a target speaker. During verification, given a test utterance, we computed the average cosine distance score between the i-vector of the test utterance and each of these $(RN + 1)$ target-speaker i-vectors. The rows labeled with "CDS+UP-AVR" in Table III show the performance of this strategy. Evidently, unlike the situation in SVM scoring, UP-AVR cannot improve the performance of cosine distance scoring. This result is reasonable because

the discriminative power of the generated (RN) i-vectors is poorer than that derived from the full-length utterances, which has detrimental effect on the average score. On the other hand, in SVM scoring, given the $RN + 1$ target-speaker's i-vectors, the SVM training algorithm can select a more relevant subset from these target-speaker's i-vectors and the background i-vectors to form a decision boundary that best discriminate the target speaker from impostors. As the SVM score is a linear weighted sum of the cosine-distance scores of these relevant (support) i-vectors and the test i-vector, each of the target-speaker's i-vectors has different contribution to the overall score and the degree of contribution is optimized by the SVM training algorithm. The aims of UP-AVR in SVM scoring is to overcome the data-imbalance problem in SVM training. Once this data-imbalance problem can be alleviated, the SVM weights can be reliably estimated.

Results in Table III also suggest that when UP-AVR is applied, a small penalty factor C is more appropriate than a large one.[5] This is reasonable because a small C leads to more target-speaker class support vectors, which improve the influence of target-speaker class data on the decision boundary of the SVMs.

VII. Conclusions

This paper applies utterance partitioning with acoustic vector resampling to i-vector speaker verification using the latest NIST SRE for performance evaluation. This work demonstrates that the approach can be effectively applied to i-vector systems in two aspects:

1) *Estimation of LDA and WCCN projection matrices*. Because a lot more i-vectors can be produced per training utterance, numerical difficulty arising from limited training sessions can be avoided. Our experimental results show that even if each training speaker has two recording sessions only, utterance partitioning can help to find more robust LDA and WCCN transformation matrices, leading to significant improvement in verification performance.
2) *SVM scoring*. It is common to use cosine distance scoring rather than SVM scoring because of the data-imbalance problem in the latter, i.e., for each speaker-dependent SVM, there is only one target-speaker's i-vector but many background-speaker i-vectors for training. This data-imbalance causes the SVM decision function to be dictated by the background-speakers' support vectors. With utterance partitioning, the data-imbalance problem in SVM scoring can be mitigated by using more target-speaker's i-vectors for training the speaker-dependent SVMs even if each target speaker provides one enrollment utterance only. Our results demonstrate that increasing the number of target-speaker's i-vectors from one i-vector per target-speaker to 9 i-vectors per target-speaker reduces the EER and minimum normalized DCF of SVM scoring by 17% and 2%, respectively. This performance improvement makes SVM scoring outperforms cosine distance scoring by 19% and 9% in terms of minimum normalized DCF and EER, respectively.

[5]As we used a scalar for the '`boxconstraint`' parameter in `svmtrain.m` provided by Mathworks, the penalty factors for speaker and impostor classes will be rescaled according to the number of training samples in these two classes.

Acknowledgment

This work was in part supported by The Hong Kong Research Grant Council, Grant No. PolyU5264/09E and PolyU Grant No. G-YJ86.

References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.
[2] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.
[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[4] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of the 9th International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, Sep. 2006, pp. 1471–1474.
[5] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.
[6] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, vol. 6, no. 1, pp. 483–502, 2005.
[7] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, pp. 1713–1726, 2000.
[8] W. Rao and M. W. Mak, "Utterance partitioning with acoustic vector resampling for i-vector based speaker verification," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, Singapore, Jun. 2012.
[9] ——, "Addressing the data-imbalance problem in kernel-based speaker verification via utterance partitioning and speaker comparison," in *Proc. of Interspeech 2011*, Florence, Aug. 2011, pp. 2717–2720.
[10] M. W. Mak and W. Rao, "Utterance partitioning with acoustic vector resampling for GMM-SVM speaker verification," *Speech Communication*, vol. 53, no. 1, pp. 119–130, Jan. 2011.
[11] "The NIST year 2010 speaker recognition evaluation plan," *http://www.itl.nist.gov/iad/mig/tests/sre/2010/index.html*.
[12] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, May 2007.
[13] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, vol. 1, Toulouse, France, May 2006, pp. 97–100.
[14] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, Jan. 2000.
[15] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. Interspeech 2009*, Sep. 2009, pp. 1559–1562.
[16] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 42–54, Jan. 2000.
[17] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, Oct. 2002.
[18] Y. M. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.
[19] H. Yu and M. W. Mak, "Comparison of voice activity detectors for interview speech in NIST speaker recognition evaluation," in *Proc. of Interspeech 2011*, Florence, Aug. 2011, pp. 2353–2356.
[20] B. Efron and G. Gong, "A leisurely look at bootstrap, the jackknife, and cross-validation," *The American Statistician*, vol. 37, no. 1, pp. 36–48, 1983.

TABLE III
THE PERFORMANCE OF I-VECTOR BASED SPEAKER VERIFICATION USING DIFFERENT SCORING METHODS.

Scoring Methods		EER(%)						MinNDCF					
		CC1	CC2	CC4	CC7	CC9	Mic	CC1	CC2	CC4	CC7	CC9	Mic
CDS		1.62	2.86	3.23	8.94	**1.71**	2.98	0.42	0.52	0.54	0.99	0.46	0.63
CDS+UP-AVR(1)		1.74	3.01	3.45	9.44	2.29	3.17	0.45	0.54	0.57	0.99	0.49	0.67
CDS+UP-AVR(2)		1.75	3.02	3.44	9.49	2.27	3.21	0.45	0.54	0.57	0.99	0.49	0.67
CDS+UP-AVR(4)		1.76	3.01	3.44	9.49	2.26	3.22	0.45	0.54	0.57	0.99	0.50	0.67
SVM	$C=1$	1.82	3.08	3.32	9.50	2.56	3.26	0.32	0.49	0.50	**0.94**	0.28	0.52
	$C=0.01$	1.85	3.26	3.47	9.49	2.56	3.31	0.32	0.48	0.46	0.99	0.36	0.55
SVM+UP-AVR(1)	$C=1$	1.54	2.86	3.17	9.50	2.24	3.04	0.28	0.47	0.46	0.95	0.23	0.49
	$C=0.01$	1.51	2.99	3.02	9.37	2.34	2.97	0.29	0.47	0.41	0.99	0.28	0.51
SVM+UP-AVR(2)	$C=1$	1.57	2.84	3.02	9.50	2.16	3.04	0.27	0.47	0.45	0.96	0.23	0.49
	$C=0.01$	1.41	2.62	**2.81**	**8.38**	2.39	2.71	0.28	0.46	**0.40**	0.99	0.32	0.51
SVM+UP-AVR(4)	$C=1$	1.54	2.85	3.10	9.47	2.17	3.03	0.27	0.47	0.45	0.95	**0.23**	**0.49**
	$C=0.01$	**1.31**	**2.62**	2.84	8.66	2.42	**2.71**	**0.28**	**0.45**	0.41	0.99	0.31	0.51

C is the SVM's penalty factor. "CC" denotes common condition. "Mic" represents all common conditions involving interview-style speech or microphone speech. "CDS": cosine distance scoring. "CDS+UP-AVR(R)": computing the average cosine distance score between the claimant's LDA+WCCN projected i-vector and $NR+1$ target-speaker's i-vectors produced by UP-AVR with number of partitions $N=4$ and number of re-sampling $R=1,2$, or 4. "SVM+UP-AVR(R)": SVM scoring with each SVM trained by using $NR+1$ target-speaker's LDA+WCCN projected i-vectors and 633 background speakers' i-vectors, where the target-speaker i-vectors were produced by UP-AVR with number of partitions $N=4$ and number of re-sampling $R=1,2$, or 4.

[21] S. Cumani and P. Laface, "Analysis of large-scale SVM training algorithms for language and speaker recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, pp. 1585–1596, Jul. 2012.

[22] S. Cumani, N. Brummer, L. Burget, and P. Laface, "Fast discriminative speaker verification in the i-vector space," in *Proc. ICASSP 2011*, Prague, Czech Republic, May 2011, p. 4852C4855.

[23] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, no. 6, pp. 1304–1312, Jun. 1974.

[24] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, Crete, Greece, Jun. 2001, pp. 213–218.

[25] "Joint factor analysis matlab demo," *http://speech.fit.vutbr.cz/software/joint-factor-analysis-matlab-demo*.

[26] W. Rao and M. Mak, "Alleviating the small sample-size problem in i-vector based speaker verification," in *Proc. of Int. Sym. on Chinese Spoken Language Processing (ISCSLP'12)*, Hong Kong, Dec 2012, pp. 335–339.

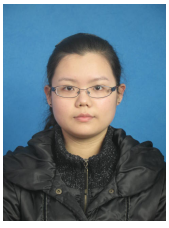
[27] K. Fukunaga, *Introduction to Statistical Pattern Classification*. Academic Press, USA, 1990.

[28] S. Raudys and R. P. W. Duin, "On expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix," *Pattern Recognition Letters*, vol. 19, pp. 385–392, 1998.

[29] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[30] W. Zhao, R. Chellappa, and P. Phillips, "Subspace linear discriminant analysis for face recognition," *Technical Report CAR-TR-914*, 1999.

[31] M. McLaren and D. van Leeuwen, "Source-normalised LDA for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, pp. 755–766, 2012.



Wei Rao received a B.Eng. degree in Electronic Information Engineering and a M.Eng. degree in Information and Telecommunication Engineering from China University of Geosciences, Wuhan, China, in 2007 and 2010, respectively. She is currently pursuing a Ph.D. degree from The Hong Kong Polytechnic University. Her current research interests include speaker recognition and machine learning.

Man-Wai Mak (M'93) received a BEng(Hons) degree in Electronic Engineering from Newcastle Upon Tyne Polytechnic in 1989 and a PhD degree in Electronic Engineering from the University of Northumbria at Newcastle in 1993. He joined the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University in 1993 and is currently an Associate Professor in the same department. He has authored more than 130 technical articles in speaker recognition, machine learning, and bioinformatics. Dr. Mak is a co-author of "Biometric Authentication: A Machine Learning Approach, Prentice Hall, 2005." He served as the Chairman of the IEEE Hong Kong Section Computer Chapter in 2003-2005 and as a Technical Committee member of the IEEE Machine Learning for Signal Processing in 2005-2007. He is currently an associate editor of IEEE Trans. on Audio, Speech and Language Processing, Journal of Signal Processing Systems, and Advances in Artificial Neural Systems. He also served as a guest editors of Journal of VLSI Signal Processing. Dr. Mak's research interests include speaker recognition, machine learning, and bioinformatics.