Utterance Partitioning with Acoustic Vector Resampling for I-Vector Based Speaker Verification

Wei RAO and Man-Wai MAK

Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University, Hong Kong SAR, China
ellen.wei-rao@connect.polyu.hk, enmwmak@polyu.edu.hk

Abstract

I-vector has become a state-of-the-art technique for text-independent speaker verification. The major advantage of i-vectors is that they can represent speaker-dependent information in a low-dimension Euclidean space, which opens up opportunity for using statistical techniques to suppress session- and channel-variability. This paper investigates the effect of varying the conversation length and the number of training sessions per speakers on the discriminative ability of i-vectors. The paper demonstrates that the amount of speaker-dependent information that an i-vector can capture will become saturated when the utterance length exceeds a certain threshold. This finding motivates us to maximize the feature representation capability of i-vectors by partitioning a long conversation into a number of sub-utterances in order to produce more i-vectors per conversation. Results on NIST 2010 SRE suggest that (1) using more i-vectors per conversation enhances the capability of LDA and WCCN in suppressing session variability, 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 SVMs to find better decision boundaries, thus making SVM scoring outperforms cosine distance scoring by 22% and 9% in terms of minimum normalized DCF and EER.

Index Terms: speaker verification, i-vectors, utterance partitioning, support vector machines.

1. Introduction

Recent research has demonstrated the merit of i-vectors [1] for text-independent speaker verification. Unlike joint factor analysis (JFA) [2] which defines two distinct subspaces: speaker space and channel space, the i-vector approach represents speakers in a single low-dimensional space named total variability space. Because this total variability space has dimension much lower than that of the GMM-supervector space, many statistical techniques such as linear discriminant analysis (LDA), within-class covariance normalization (WCCN) [3], and probabilistic LDA [4] can be applied to suppress the channel- and session-variability.

Both LDA and WCCN involve the computation of a within-class covariance matrix that requires many speakers with multiple sessions per speaker. For best performance and numerical stability, one should opt for a large number of sessions per speaker. But in practice, it is costly and inconvenient to collect such a corpus. 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. The lack of multiple sessions per speaker could result in incomplete within speaker scatter matrix [5]. This paper aims to investigate the effect of varying the number of sessions per speaker on the LDA and WCCN projection matrices and how the lack of multiple speaker sessions degrades the verification performance.

The idea of i-vectors is 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-dim 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 investigated the relationship between the length of the utterances and the discriminative power (Fisher discriminate ratio) of the resulting i-vectors. Interestingly, we observed that the discriminative power of the i-vectors becomes saturated quickly and flatten out when the utterances exceed 2–3 minutes.

The above finding motivates us to divide a long utterance into a number of sub-utterances so that multiple i-vectors can be produced for each utterance. We applied our recently proposed Utterance Partitioning with Acoustic Vector Resampling (UP-AVR) [6] to perform the partitioning. The idea is to produce a desirable number of i-vectors for each long utterance without significantly reducing the feature representation power of the i-vectors. This is achieved by randomly selecting a subset of acoustic vectors from the full-length acoustic vector sequence for estimating an i-vector. This resampling procedure is repeated several times to produce a desirable number of i-vectors.

It turns out that this partitioning technique is beneficial to (1) the estimation of LDA and WCCN projection matrices and (2) SVM scoring based on the projected i-vectors. For the former, because a lot more i-vectors can be produced per training speaker, numerical stability problems can be avoided even if only two sessions per speaker are available. Using the interview and microphone speech in NIST 2008 SRE data for training the LDA and WCCN projection matrices and NIST 2010 SRE data for evaluation, we observed that when each training speaker has five recording sessions, UP-AVR can reduce the EER from 12.21% to 5.55%. For the latter, it is common to compute the cosine distance scores [1] of the LDA+WCCN projected vectors. The use of SVM scoring is not preferred in the literature because of the data-imbalance problem, 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

This work was in part supported by The Hong Kong Research Grant Council, Grant No. PolyU5264/09E and PolyU Grant No. 4-ZZ7W.

data-imbalance causes the SVM decision function to be dictated by the background-speakers' support vectors [7]. However, with the UP-AVR, this data-imbalance problem in SVM scoring can be readily mitigated by using more target-speaker's i-vectors for training the speaker-dependent SVMs. This paper demonstrates that with UP-AVR, SVM scoring can outperform cosine distance scoring by 22% and 9% in terms of minimum DCF and EER, respectively.

The paper is organized as follows. Section 2 outlines the i-vector framework for speaker verification, followed by an experiment highlighting the relationship between the utterance length and the discriminative power of LDA+WCCN projected i-vectors. Sections 4 describes the idea of UP-AVR and its applications to the i-vector framework. In Sections 5 and 6, we report evaluations based on NIST 2010 SRE. Section 7 concludes the findings.

2. The I-vector Framework For Speaker Verification

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

2.1. I-vector Extraction

The i-vector approach is based on the idea of joint factor analysis (JFA) [8]. 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 [9] m_s is written as:

$$m_s = m + Tw, \tag{1}$$

where m is the GMM-supervector of the universal background model (UBM) [10] which is speaker- and channel- independent, T is a low-rank total variability matrix, and w is a low-dimension 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.

2.2. 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.

2.2.1. *Linear Discriminant Analysis*

Linear discriminant analysis (LDA) is a commonly used technique for dimensionality reduction. The idea of this approach is to find a set of orthogonal axes for minimizing the within-class variation and maximizing the between-class variation. In the i-vector framework, the i-vectors of a speaker constitute a class, leading to the following objective function for multi-class LDA [11]:

$$J(A) = \text{tr}\left\{ \left(A^{\mathsf{T}} S_w A\right)^{-1} \left(A^{\mathsf{T}} S_b A\right) \right\} \tag{2}$$

where A comprises the optimal directions on which the i-vectors should be projected, S_w is the within-speaker scatter matrix, and S_b is the between-class scatter matrix. These two scatter matrices are written as follows:

$$S_w = \sum_{i=1}^{S} \frac{1}{M_i} \sum_{j=1}^{M_i} (w_j^i - \mu^i)(w_j^i - \mu^i)^{\mathsf{T}} \tag{3}$$

and

$$S_b = \sum_{i=1}^{S} (\mu^i - \mu)(\mu^i - \mu)^{\mathsf{T}}, \tag{4}$$

where

$$\mu^i = \frac{1}{M_i} \sum_{j=1}^{M_i} w_j^i, \tag{5}$$

μ^i 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 μ is the global mean of all i-vectors in the training dataset. Maximizing Eq. 2 leads to the projection matrix A that comprises the leading eigenvectors of $S_w^{-1} S_b$.

2.2.2. *Within Class Covariance Normalization*

Within Class Covariance Normalization (WCCN) [3] 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:

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

$$\widetilde{\mu^i} = \frac{1}{M_i} \sum_{j=1}^{M_i} A^{\mathsf{T}} w_j^i, \tag{7}$$

where A is the LDA projection matrix. The WCCN projection matrix B can be obtained by Cholesky decomposition of $W^{-1} = BB^{\mathsf{T}}$.

2.3. Scoring Methods

2.3.1. *Cosine Distance Scoring*

Cosine distance scoring (CDS) [12] 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 ($w^{(c)}$) and target-speaker's i-vector ($w^{(s)}$) in the LDA+WCCN projection space:

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

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

2.3.2. *Support Vector Machine Scoring*

The idea of support vector Machine (SVM) scoring in i-vector speaker verification [12] 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 the i-vectors 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

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

where $\alpha_0^{(s)}$ is the Lagrange multiplier corresponding to the target speaker,[1] $\alpha_i^{(s)}$'s are Lagrange multipliers corresponding to the background speakers, $\mathcal{S}^{(b)}$ is a set containing the indexes of the i-vectors in the background-speaker set, and $w^{(b_i)}$ 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. It was found [1] that the cosine kernel is appropriate. Specifically,

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

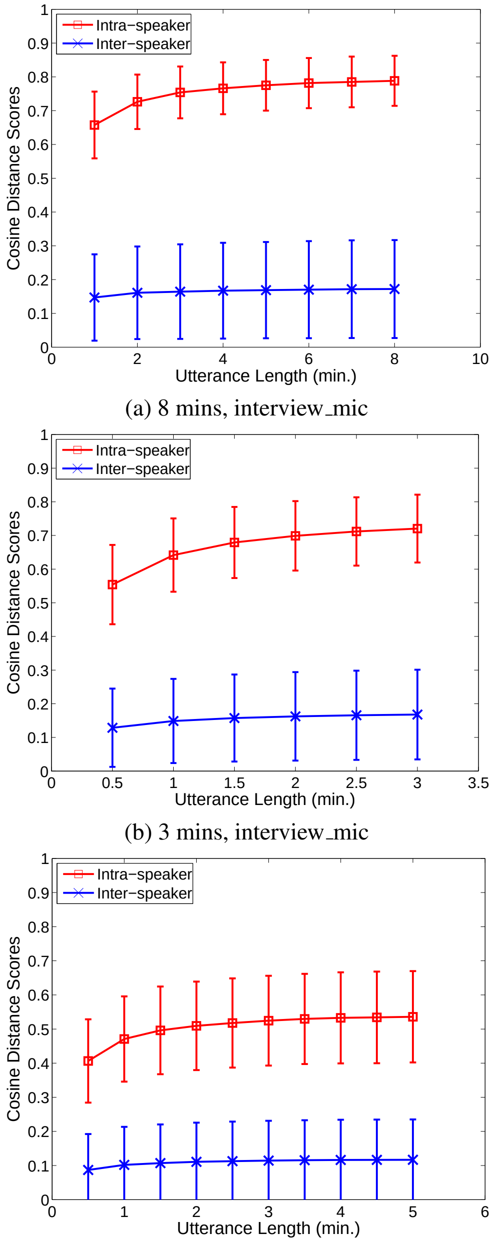
where we replace $w^{(s)}$ by $w^{(b_i)}$ for evaluating the second term of Eq. 9. Note that Eq. 8 and Eq. 10 are the same. However, their role in the scoring process is different. The former is directly used for calculating the score, whereas the latter is used for kernel evaluation.

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. We have recently proposed a method called utterance partitioning that can alleviate this problem, which will be described in details in Section 4.

3. 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, phonecall_mic, and phonecall_tel sessions of NIST 2010 SRE. For each conversation, VAD [13] is first applied to extract the speech segments, followed by partitioning the segments into equal-length sub-utterances. Then, variable numbers of sub-utterances were packed to estimate the i-vectors, followed by LDA and WCCN projections to 150-dim vectors. Cosine-distance scores were obtained from these 150-dim vectors.

Figure 1 shows the mean intra- and inter-speaker scores (with error bars indicating one standard deviation) of the three types of speech. Apparently, 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



(a) 8 mins, interview_mic

(b) 3 mins, interview_mic

(c) 5 mins, phonecall_tel

Figure 1: Intra-speaker and inter-speaker cosine-distance scores versus utterance length. For "8 mins, 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 12992 inter-speaker scores for each utterance length. For "3 mins, interview_mic", the scores were obtained from the 3-min interview sessions of 196 male speakers, each providing 4 interview conversations. This amounts to 1176 intra-speaker scores and 611,520 inter-speaker scores for each utterance length. For "5 mins, phonecall_tel", the scores were obtained from the 5-min phonecall conversations of 47 male speakers, each providing 4 conversations. This amounts to 174 intra-speaker scores and 34,592 inter-speaker scores for each utterance length. For each conversation, VAD [13] was first applied to extract the speech segments, followed by partitioning the segments into equal-length sub-utterances. Then, variable numbers of sub-utterances were packed to estimate the i-vectors, followed by LDA and WCCN projections to 150-dim vectors. Cosine-distance scores were obtained from these 150-dim vectors.

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

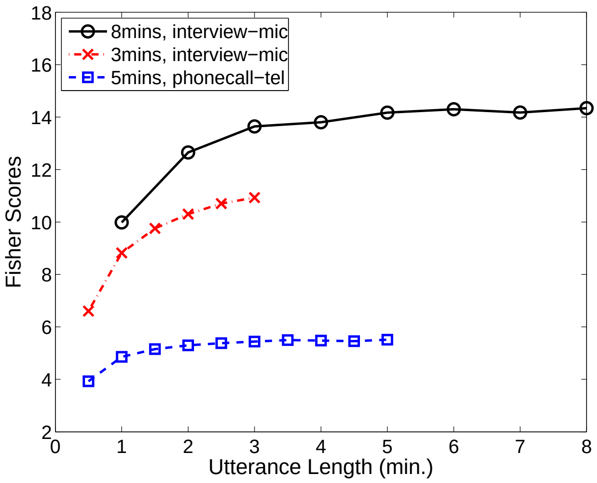


Figure 2: Fisher discriminant ratio (Eq. 11) derived from the intra- and inter-speaker cosine-distances versus utterance length. See the caption of Figure 1 for the details of intra- and inter-speaker distance.

of the i-vectors with respect to the segment length, we plot in Figure 2 the Fisher discriminant ratio

$$S_{\text{Fisher}} = (\mu_{\text{intra}} - \mu_{\text{inter}})^2/(\sigma^2_{\text{intra}} + \sigma^2_{\text{inter}}) \qquad (11)$$

between the intra- and inter-speaker scores whose mean and standard deviation are respectively denoted by μ and σ. The larger the Fisher discriminant ratio, the higher the discriminative power. The result clearly suggests that the discriminative power becomes saturated for segment length exceeding 2–3 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 recordings are already available, it may be beneficial to divide the long utterances 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.

4. Utterance Partitioning with Acoustic Vector Resampling

Utterance partitioning with acoustic vector resampling (UP-AVR) [7] 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 partitions an enrollment utterance into a number of sub-utterances, with each segment producing one i-vector. 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. To produce a sufficient number of sub-utterances without compromising their representation power, UP-AVR uses the notion of random resampling in bootstrapping [14]. The idea is based on the fact that changing the order of acoustic vectors will not affect the resulting i-vector. 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.

UP-AVR was originally introduced to alleviate the the data imbalance problem in GMM-SVM [9]. In the current work, we found that UP-AVR is also applicable to the i-vector framework. First, it can improve the effectiveness of LDA and WCCN under limited speech resources. Second, it can be applied to alleviate the data imbalance problem in SVM scoring.

4.1. UP-AVR for LDA and WCCN

The aim of LDA is to find a set of axes that minimize the intra-speaker variation and maximize the inter-speaker variation. It requires a sufficient number of recording sessions per training speaker for estimating the inter- and intra-speaker covariance matrices. However, collecting such recordings is costly and inconvenient. As demonstrated in Section 3, when the utterance length for i-vector extraction is sufficiently long, further increasing the length will not increase the i-vectors' discriminative power significantly. Therefore, given a long utterance, some intrinsic speaker information will be wasted if the whole utterance is used for estimating the i-vector. To make a better use of the long utterance, we can apply UP-AVR to partition the utterance so that more i-vectors can be produced for estimating the LDA and WCCN projection matrix. It not only solves the numerical problem caused by insufficient data for LDA, but also reduces the intra-speaker variation.

4.2. UP-AVR for SVM Scoring

A strategy for solving the data imbalance problem in SVM scoring is to increase the number of minority-class samples for training the SVMs. One may use more enrollment utterances, which means more i-vectors from the speaker class. However, this strategy shifts the burden to the users by requesting them to provide multiple enrollment utterances, which may not be practical. Through UP-AVR, many sub-utterances of a target speaker can be generated based on his/her original utterance and each sub-utterance can produce an i-vector, which improves the influence of target-speaker class data on the decision boundary of the SVM.

5. Experiments

5.1. 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. 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. Silence regions of the utterances in these corpora were removed by a VAD [13]. Cepstral mean normalization [15] was then applied to the MFCCs, followed by feature warping [16] using a window of 3 seconds. 19 MFCCs plus their 1st- and 2nd- derivatives were extracted from the speech regions of each utterance, leading to 60-dim acoustic vectors.

5.2. Total Variability Modeling and Channel Compensation

The i-vector systems are based on a gender-dependent UBM with 1024 mixtures. We selected 6,102 utterances from 192 speakers (each with at least 8 utterances) in NIST 2005–2008

SRE to estimate a total variability matrix with 400 total factors. A modified version of the BUT JFA Matlab code was used for i-vector training and scoring. Before calculating the verification scores, LDA and WCCN projections were performed for channel compensation. We used the same data set for training the total variability matrix to estimate the LDA and WCCN matrices. After LDA and WCCN projections, the dimension of i-vectors was reduced to 150.

5.3. Scoring Method and Score Normalization

In this paper, we adopted two scoring method: SVM scoring and cosine distance scoring. For building the SVM classifier, we selected 633 impostors from NIST 2005–08 SREs. ZT-norm [17] was used for score normalization. 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 2005–08 SREs.

6. Results and Discussions

6.1. UP-AVR for LDA and WCCN

The purpose of this experiment is to investigate the performance of i-vector based systems under insufficient data for training the LDA and WCCN matrices. Therefore, we only used the interview and microphone speech of NIST 2008 SRE for training the matrices. We started from not using intersession compensation, i.e. without applying LDA and WCCN on the i-vectors. Then, we progressively increased the number of recording sessions per training speakers for training the LDA and WCCN projection matrices. Table 1 shows the effect of varying the number of recordings per speaker on the effectiveness of LDA and WCCN projection. Without channel compensation (System A), the performance is very poor. The results of System B suggest that insufficient number of recordings per speaker can lead to inaccurate projection matrices, causing the performance even poorer than the one without applying LDA and WCCN (System A). This observation also agrees with the findings in [5]. By increasing the number of recordings per speaker, the performance of the i-vector systems improves significantly.

Table 2 shows the overall performance of interview and microphone speech, which was obtained by concatenating the scores of Common Conditions 1, 2, 4, 7, and 9 in NIST 2010 SRE. When each training speaker has less than 5 utterances, numerical difficulty occurs while training the LDA and WCCN matrices.[2] Even if the number of recordings per training speaker increases to 5, it is still insufficient to estimate the projection matrices. On the other hand, if UP-AVR is applied to increase the number of i-vectors per training speaker, the performance improves significantly. Although UP-AVR generates the sub-utterances from the utterance in the same recording session, it can help LDA and WCCN minimizing the intra-speaker variation. However, 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 experiment).

6.2. UP-AVR for SVM Scoring

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

	No. of utts. per speaker (M)					
Systems	2	3	4	5	6	≥ 8
Without UP-AVR	–	–	–	0.99	0.86	0.74
UP-AVR(2)	0.97	0.94	0.85	**0.80**	0.81	0.74
UP-AVR(4)	0.94	0.91	0.85	0.84	**0.80**	**0.74**
UP-AVR(8)	**0.91**	**0.91**	**0.85**	0.82	0.83	0.75

(a) MinNDCF

	No. of utts. per speaker (M)					
Systems	2	3	4	5	6	≥ 8
Without UP-AVR	–	–	–	12.21	7.76	3.96
UP-AVR(2)	15.24	8.97	6.71	6.21	6.18	3.83
UP-AVR(4)	9.51	7.28	**5.74**	5.68	5.61	3.74
UP-AVR(8)	**8.13**	**6.93**	6.34	**5.55**	**5.58**	**3.67**

(b) EER(%)

Table 2: The performance of i-vector based speaker verification with and without partitioning the full-length utterances for training the LDA and WCCN projection matrices. M is the number of utterances per speaker used for training the matrices, and $M \geq 8$ means at least 8 utterances per speaker were used for training. UP-AVR(N) means dividing the full-length training utterances (obtained from the microphone speech of 111 speakers in NIST 2008 SRE) into N partitions using UP-AVR. In all cases, the number of re-sampling in UP-AVR is set to 1, i.e. $R = 1$. "–" denotes the situation where numerical difficulty occurs when estimating the projection matrices.

Table 3 compares the performance between SVM scoring and cosine distance scoring in i-vector based speaker verification. Table 3 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, SVM scoring can outperform cosine distance scoring by 22% and 9% in terms of minimum DCF and EER, respectively. Results in Table 3 also suggest that when UP-AVR is applied, a small penalty factor C is more appropriate than a large one. 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.

7. 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 is not only effective in overcoming the data imbalance problem in SVM scoring but also able to improve the effectiveness of LDA and WCCN projections under insufficient speech resources for training these projection matrices.

8. 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. Du-

[2] Maximizing Eq. 2 leads to the LDA projection matrix A that comprises the leading eigenvectors of $S_w^{-1}S_b$. However, insufficient data will cause S_w closed to singular.

No. of utts. per speaker (M)	MinNDCF						EER (%)					
	CC1	CC2	CC4	CC7	CC9	Mic	CC1	CC2	CC4	CC7	CC9	Mic
(A) $M = 0$ (Without LDA and WCCN)	0.62	0.84	0.82	0.98	0.63	0.90	5.10	10.77	9.65	15.08	5.96	12.60
(B) $M = 5$	0.97	0.99	0.97	0.99	0.98	0.99	8.52	12.75	11.72	18.43	10.25	12.21
(C) $M = 6$	0.76	0.86	0.79	0.97	0.56	0.86	4.50	8.52	6.10	12.84	5.13	7.76
(D) $M = 7$	0.71	0.81	0.82	**0.96**	0.56	0.85	3.83	6.84	5.19	12.28	4.27	6.53
(E) $M \geq 8$	**0.54**	**0.64**	**0.72**	0.98	**0.42**	**0.74**	**2.43**	**3.79**	**3.85**	**10.87**	**4.09**	**3.96**

Table 1: The performance of i-vector based speaker verification using different numbers of recordings (utterances) per speaker for training the LDA and WCCN matrices. 111 male speakers were selected from NIST 2008 SRE, each speaker provides an average of 32 utterances recorded by various types of microphones. "CC" denotes common condition. "Mic" represents all common conditions involving interview-style speech or microphone speech. $M = 0$ means without applying LDA and WCCN. $M = x$ 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 32 recordings per speaker.

Scoring Methods		Common Condition					
		CC1	CC2	CC4	CC7	CC9	Mic
CDS		0.38	0.52	0.53	0.99	0.44	0.63
SVM	$C = 1$	0.33	0.49	0.50	0.91	0.28	0.52
	$C = 0.01$	0.30	0.47	0.45	0.99	0.29	0.53
SVM+UP-AVR(4)	$C = 1$	0.29	0.47	0.46	0.92	0.29	0.49
	$C = 0.01$	0.28	0.45	0.41	0.99	**0.23**	0.49
SVM+UP-AVR(16)	$C = 1$	0.30	0.46	0.47	**0.89**	0.24	0.49
	$C = 0.01$	**0.26**	**0.44**	**0.41**	0.99	0.24	**0.49**

(a) MinNDCF

Scoring Methods		Common Condition					
		CC1	CC2	CC4	CC7	CC9	Mic
CDS		1.72	2.88	2.81	9.23	**1.71**	2.98
SVM	$C = 1$	1.87	3.12	3.06	10.05	2.56	3.26
	$C = 0.01$	1.87	3.30	3.07	8.94	3.31	3.35
SVM+UP-AVR(4)	$C = 1$	1.71	3.00	2.86	10.05	2.56	3.10
	$C = 0.01$	1.57	3.04	2.97	8.37	3.04	3.07
SVM+UP-AVR(16)	$C = 1$	1.64	3.03	2.79	9.31	2.56	3.12
	$C = 0.01$	**1.46**	**2.76**	**2.70**	**8.30**	3.18	**2.73**

(b) EER(%)

Table 3: The performance of i-vector based speaker verification using different scoring methods. C is the user-defined penalty parameter for training the SVMs; *CDS*: cosine distance scoring; *SVM*: SVM scoring with each SVM trained by using one LDA+WCCN projected i-vector from a target speaker and 633 i-vectors from background speakers; *SVM+UP-AVR(4)*: SVM scoring with each SVM trained by using 5 target-speaker's LDA+WCCN projected i-vectors and 633 background speakers' i-vectors, each i-vector derived from a sub-utterance produced by UP-AVR with $N = 4$ and $R = 1$; *SVM+UP-AVR(16)*: SVM scoring with each SVM trained by using 17 target-speaker's LDA+WCCN projected i-vectors and 633 background speakers' i-vectors, each i-vector derived from a sub-utterance produced by UP-AVR with $N = 4$ and $R = 4$.

mouchel, "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] 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.

[4] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey2010 - The Speaker and Language Recognition Workshop*, Brno, Czech Republic, 2010.

[5] 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.

[6] W. Rao and M.W. Mak, "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.

[7] 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.

[8] 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.

[9] 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*, Toulouse, France, May 2006, vol. 1, pp. 97–100.

[10] 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.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[12] 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.

[13] H.B. 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.

[14] 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.

[15] 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.

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

[17] 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.