Acoustic Vector Resampling for GMMSVM-Based Speaker Verification

Man-Wai MAK and Wei RAO

Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University

Abstract

Using GMM-supervectors as the input to SVM classifiers (namely, GMM-SVM) is one of the promising approaches to text-independent speaker verification. However, one unaddressed issue of this approach is the severe imbalance between the numbers of speaker-class utterances and impostor-class utterances available for training a speaker-dependent SVM. This paper proposes a resampling technique – namely utterance partitioning with acoustic vector resampling (UP-AVR) – to mitigate the data imbalance problem. Specifically, the sequence order of acoustic vectors in an enrollment utterance is first randomized; then the randomized sequence is partitioned into a number of segments. Each of these segments is then used to produce a GMM-supervector via MAP adaptation and mean vector concatenation. A desirable number of speaker-class supervectors can be produced by repeating this randomization and partitioning process a number of times. Experimental evaluations suggest that UP-AVR can reduce the EER of GMM-SVM systems by about 10%.

1. Introduction

Recent research has demonstrated the merit of support-vector-machine (SVM) scoring in text-independent speaker verification. This approach derives a GMM-supervector [1] by stacking the mean vectors of a target-speaker dependent, MAP-adapted GMM [2]. The supervector is then presented to a speaker-dependent SVM for scoring, leading to the GMM-SVM framework. 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 [3]. SVM scoring can also be used in the joint factor analysis (JFA) framework, where speaker factors are used as input to the SVMs [4].

A problem in SVM scoring is that the number of target speaker utterances for training the target-speaker's SVM is very limited (typically only one enrollment utterance is available). Given that the number of background speakers' utterances is typically several hundreds, the limited number of enrollment utterances leads to a severe data imbalance problem. An undesirable consequence of data imbalance is that the orientation of the decision boundary is largely dictated by the data in the majority (background speakers) class.

Approaches to mitigating the effect of imbalanced data on SVM classifiers can be divided into data processing approaches and algorithmic approaches. The former attempts to re-balance the training data without changing the SVM training algorithm. This category can be further divided into (1) over-sampling [5] where more positive (minority class) training examples are generated from existing data, (2) under-sampling [6] where a subset of negative (majority class) training samples are selected for each entity in an ensemble of SVMs, and (3) combinations of over- and under-sampling [6]. The algorithmic approaches attempt to modify the training algorithms to mitigate the effect caused by data imbalance, e.g., assigning different error cost to positive and negative training samples [7] or modifying the kernel according to the distribution of training data [8].

Unlike many other problem domains, the data imbalance problem in GMM-SVM speaker verification is special in that the number of minority-class samples (enrollment utterances) is extremely small. In fact, it is not uncommon to have only one enrollment utterance per client speaker. This extreme data imbalance excludes the use of over-sampling methods such as SMOTE [5] where minority-class samples are generated based on the existence of some (but not one) minority-class samples. Under-sampling is also not an option, because of information loss that arise from discarding important background speakers.

Obviously, the data-imbalance problem can be tackled by requesting target speakers to provide speech in multiple sessions, preferably with different acoustic conditions. However, this strategy simply shifts the burden to the target speakers. A better strategy is to derive a training algorithm that makes the best use of the single-session utterance. In this paper, we look at over-sampling in another dimension. Instead of creating more minority-class samples from existing ones, we generate minority-class samples by partitioning the sequence of acoustic vectors in the enrollment utterance into a number of segments or sub-utterances, with each segment producing one GMM-supervector. To increase the number of segments, one may reduce the length of sub-utterances. However, this will inevitably compromise the representation power of the sub-utterances. Here, we propose randomizing the sequence order before partitioning takes place. This randomization and partitioning process can be repeated several times to produce a desirable number of GMM-supervectors. The randomization process ensures that the GMM-supervectors are different from repetition to repetition.

2. GMM-SVM

The idea of GMM-SVM [1] is to harness the discriminative information embedded in the training data by constructing an SVM that optimally separates the GMM of a target speaker from the GMMs of background speakers. Given the SVM of target speaker s, the verification score of $\text{utt}^{(c)}$ is given by

$$S_{\text{GMM-SVM}}(\text{utt}^{(c)}) = \alpha_0^{(s)} K\left(\text{utt}^{(c)}, \text{utt}^{(s)}\right) - \sum_{i \in \mathcal{S}^{(b)}} \alpha_i^{(s)} K\left(\text{utt}^{(c)}, \text{utt}^{(b_i)}\right) + d^{(s)}, \quad (1)$$

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

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 support vectors in the background-speaker set, and $\text{utt}^{(b_i)}$ is the utterance of the i-th background speaker. Note that only those background speakers with non-zero Lagrange multipliers have contribution to the score. The kernel function $K(\cdot,\cdot)$ can be of many forms. The most common being the Mahalanobis kernel (also called GMM-supervector kernel) [1]:

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

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

3. Utterance Partitioning for GMM-SVM

In typical GMM-SVM setting, there is only one speaker-class's supervector for training. The problem is that the SVM decision plane is largely governed by the impostor-class supervectors (support vectors). As depicted in Fig. 1, there is a region in the supervector space where the speaker-class supervector can move around without affecting the orientation of the decision planes.

To increase the influence of speaker-class data on the decision plane, this paper proposes partitioning an enrollment utterance into a number of sub-utterances. To produce a sufficient number of sub-utterances without compromising their representation power, we use the notion of random resampling in bootstrapping [9]. The idea is based on the fact that MAP adaptation uses the statistics of the whole utterance to update the GMM parameters. In other words, changing the order of acoustic vectors will not affect the resulting MAP-adapted model. Therefore, we may randomly rearrange the acoustic vectors in an utterance and then partition the utterance into N sub-utterances and repeat the process as many times as appropriate. More precisely, if this process is repeated R times, we obtain RN sub-utterances from a single enrollment utterance. We refer to this approach as utterance partitioning with acoustic vector resampling (UP-AVR). Its procedure is as follows:

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

Step 2: For each background speaker, use his/her N sub-utterances and full-length utterance to create $N+1$ background GMM-supervectors. For B background speakers, this procedure results in $B(N+1)$ background supervectors.

Step 3: Given an enrollment utterance of a target speaker, compute its acoustic vectors and randomize their sequence of occurrences in the utterance. Divide the randomized sequence of acoustic vectors into N partitions (subsequences). Use the N sub-sequences to create N GMM-supervectors by adapting the UBM.

Step 4: Repeat Step 3 R times to obtain RN target speaker's supervectors; together with the full-length utterance, form $RN+1$ speaker's supervectors.

Step 5: Use the $RN+1$ supervectors created in Steps 3 and 4 as positive-class data and the $B(N+1)$ background supervectors created in Step 2 as negative-class data to train a linear SVM for the corresponding target speaker.

The same partitioning strategy are applied to both target-speaker utterances and background utterances so that the length of target-speaker's sub-utterances matches that of the background speakers' sub-utterances. Matching the duration of target-speaker utterances with that of background utterances has been found useful in previous studies [10].

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

Fig. 2 further demonstrates the benefit of increasing the number of speaker-class supervectors. In the figure, the larger the difference between the speaker-class scores and impostor-class scores (Eq. 1), the larger the discriminative power of the SVM. As expected, the SVM trained with 5 target-speaker utterances [(A), green] exhibits the greatest discriminative power and the largest score difference (1.37). However, this strategy uses 5 times as much speech materials as using one target-speaker utterance [(B), blue-dashed] for training. Evidently, the discriminative power of the SVM trained with the speaker-class supervectors generated by UP-AVR [(D)–(F)] is greater than that trained with only one speaker-class supervector (B). The overlapping between (B) and (C) suggests that synthesizing speaker-class supervectors in the neighborhood of the speaker-class supervector has little effect on the decision boundary, resulting in almost no change in the SVM scores. This agrees with the hypothetical situation shown in Fig. 1, where changing the speaker-class supervectors within a certain region will not change the decision plane. The overlapping between (E) and (F) suggests that just 5 speaker-class supervectors is sufficient, which agrees with the results in Table 3 and Figure 3.

4. Experiments

NIST 1999–2002 Speaker Recognition Evaluation (SRE), NIST 2004 SRE (1side-1side), and Fisher were used in the experiments. NIST'99–01 SRE and Fisher were used as development data (NAP, UBM, and T-norm), and NIST'02 and NIST'04 were used for performance evaluations (see Table 1 for details).

For each utterance, silence regions were removed by an energy-based VAD. Twelfth-order MFCCs plus their first derivative were extracted from the speech regions of the utterance. Cepstral mean normalization was applied to the MFCCs, followed by feature warping [11]. Then, UP-AVR was applied to the feature vectors of each utterance.

For GMM-UBM, the number of mixtures is set to 1,024. MAP adaptation [2] (relevance factor = 16) was used to create the speaker and Tnorm models. For GMM-SVM, the number of mixtures was empirically optimized in NIST'02 and the resulting value was applied to NIST'04. It was found that 256 leads to the lowest EER and min DCF. This result also agrees with the results in [12]. NAP [13] was applied to all GMM-supervectors, followed by SVM scoring and T-norm.

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

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

Table 1: The roles of different corpora in the performance evaluations.

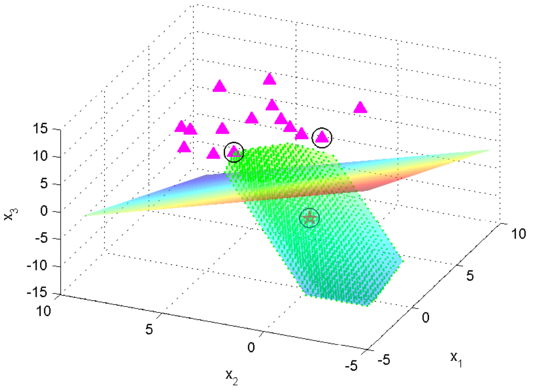

Figure 1: A 3-dim two-class problem illustrating the imbalance between the number of minority-class samples (red pentagram) and majority-class samples (pink triangles) in a linear SVM. The decision plane is defined by 3 support vectors (enclosed by black circles). The green region beneath the decision plane represents the region where the minority-class sample can be located without changing the orientation of the decision plane.

5. Results and Discussions

Figure 3 shows the trend of EER when the number of speaker-class supervector increases. It shows that utterance partitioning can reduce EER (also minimum DCF, but result not shown here). More importantly, the most significant performance gain is obtained when the number of speaker-class supervectors increases from 1 to 5, and the performance levels off when more supervectors are added. This is reasonable because a large number of positive supervectors will only result in a large number of zero Lagrange multipliers for the speaker class and increase the correlation among the synthesized supervectors.

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

Table 2 shows the EER and minimum DCF in NIST'02. The results clearly demonstrate the merit of utterance partitioning, particularly the one with acoustic vector resampling.

Table 3 shows the performance of UP-AVR in NIST'04 when the number of speaker-class supervectors increases from 5 to 201. The results suggest that with just 5 speaker-class supervectors (UP-AVR(5)), significant reduction in EER can be obtained (p-value of McNemar's test < 0.005). However, adding extra speaker-class supervectors can only reduce the EER slightly, which agrees with the SVM scores in Fig. 2 and confirms our earlier argument that it is not necessary to generate an excessive number of speaker-class supervectors. The p-value of McNemar's test between Systems C and D in Table 3 is less than 0.005. Because the EER of System D is higher than other systems that use UP-AVR, we conclude that all of the systems that use UP-AVR are significantly better than the one without using UP-AVR.

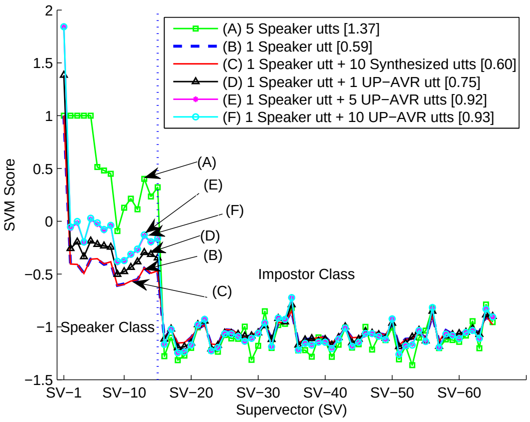

Figure 2: Scores produced by SVMs that use one or more speaker-class supervectors (SVs) and 250 background SVs for training. The horizontal axis represents the training/testing SVs. Specifically, SV-1 to SV-15 were obtained from 15 utterances of the same target-speaker, and SV-16 to SV-65 were obtained from the utterances of 50 impostors. For (B) to (F) in the legend, SV-1 was used for training and SV-2 to SV-15 were used as speaker-class test vectors, i.e., the score of SV-1 is the training-speaker score, whereas the scores of SV-2 to SV-15 are test speaker scores. For (A), SV-1 to SV-5 were used for training and SV-6 to SV-15 were used for testing. In (C), 10 speaker-class SVs were generated by a Gaussian generator using SV-1 as the mean vector and component-wise intra-speaker variances as the diagonal covariance matrix; whereas in (D)–(F), 1, 5, and 10 speaker-class UP-AVR SVs were obtained from the utterance corresponding to SV-1. Values inside the squared brackets are the mean difference between speaker scores and impostor scores. NAP have been applied to the SVs.

6. Conclusion

This paper has proposed an approach to increasing the number of target-speaker's supervectors in GMM-SVM speaker verifi-

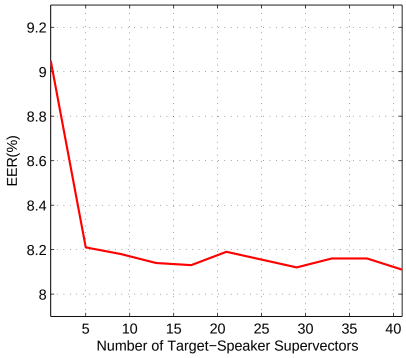

Figure 3: EER versus the number of speaker-class supervectors used for training the speaker-dependent SVMs in NIST'02. The supervectors were obtained by using UP-AVR with $N = 4$ and different values of R.

Method	EER(%)	MinDCF
(A) GMM-UBM+TNorm	10.29	0.0428
(B) GMM-SVM+NAP+TNorm	9.05	0.0362
(C) GMM-SVM+NAP+TNorm+UP(5)	8.46	0.0342
(D) GMM-SVM+NAP+TNorm+UP-AVR(33)	8.16	0.0337

Table 2: Performance of GMM-UBM, GMM-SVM, and GMM-SVM with utterance partitioning in NIST'02. The numbers inside the parentheses indicate the number of speaker-class supervectors used for training a speaker-dependent SVM, which include the supervectors generated by UP-AVR ($N = 4$, $R = 8$) and the full-length utterance. The p-values of McNemar's test between Systems (B) and (D) is less than 0.005.

cation. The paper demonstrates that a useful set of speaker-class supervectors can be generated by randomizing the sequence order of acoustic vectors in an enrollment utterance, followed by partitioning the randomized acoustic vectors. Evaluations show that the generated supervectors can alleviate the data imbalance problem and help the SVM learning algorithm to find better decision boundaries, thereby improving the verification performance. The proposed resampling technique has important implications to practical implementation of speaker verification systems because it reduces the number of enrollment utterances and thereby reducing the burden and time users spent on speech recording.

Method	EER	MinDCF
(A) GMM-UBM+TNorm	16.05	0.0601
(B) GMM-SVM+TNorm	13.40	0.0516
(C) GMM-SVM+NAP+TNorm	10.42	0.0458
(D) GMM-SVM+NAP+TNorm+UP-AVR(5)	9.67	0.0421
(E) GMM-SVM+NAP+TNorm+UP-AVR(61)	9.63	0.0422
(F) GMM-SVM+NAP+TNorm+UP-AVR(101)	9.46	0.0419
(G) GMM-SVM+NAP+TNorm+UP-AVR(201)	9.58	0.0421

Table 3: Performance of GMM-UBM, GMM-SVM, and GMM-SVM with utterance partitioning in NIST'04 (core test, all trials). The numbers inside the parentheses indicate the number of speaker-class supervectors used for training a speaker-dependent SVM, which include the supervectors generated by UP-AVR ($N = 4$, $R = 1, 15, 25, 50$) and the full-length utterance. The p-values of McNemar's tests between Systems D-G and System C are less than 0.005.

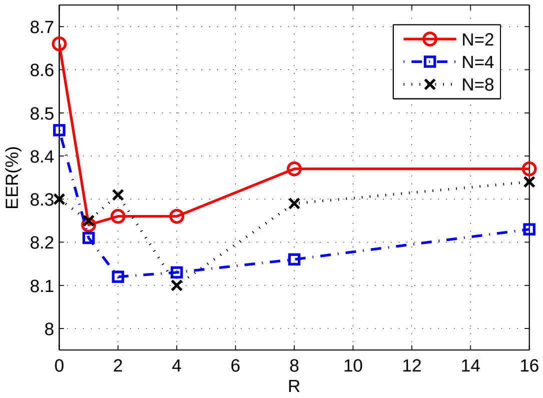

Figure 4: Performance of UP-AVR for different numbers of partitions (N) and resampling (R) in NIST'02. When $R = 0$, UP-AVR is reduced to UP.

7. References

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

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

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

[4] N. Dehak, et al., "Support vector machines and joint factor analysis for speaker verification," in *ICASSP*, 2009, pp. 4237–4240.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Artificial Intelligence and Research*, vol. 16, pp. 321V357, 2002.

[6] Y. Tang, Y.Q. Zhang, N.V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. on System, Man, and Cybernetics, Part B*, vol. 39, no. 1, pp. 281–288, Feb 2009.

[7] Y. Lin, Y. Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, no. 1-3, pp. 191–202, 2002.

[8] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[9] 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.

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

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

[12] M. Ferras, C. C. Leung, C. Barras, and J. L. Gauvain, "Comparison of speaker adaptation methods as feature extraction for SVM-based speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2009.

[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'06*, 2006, vol. 1, pp. 97–100.