

Addressing the Data-Imbalance Problem in Kernel-based Speaker Verification via Utterance Partitioning and Speaker Comparison

Wei RAO and Man-Wai MAK

Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University

Abstract

GMM-SVM has become a promising approach to text-independent speaker verification. However, a problematic issue of this approach is the extremely serious imbalance between the numbers of speaker-class and impostor-class utterances available for training the speaker-dependent SVMs. This data-imbalance problem can be addressed by (1) creating more speaker-class supervectors for SVM training through utterance partitioning with acoustic vector resampling (UP-AVR) and (2) avoiding the SVM training so that speaker scores are formulated as an inner product discriminant function (IPDF) between the target-speaker's supervector and test supervector. This paper highlights the differences between these two approaches and compares the effect of using different kernels – including the KL divergence kernel, GMM-UBM mean interval (GUMI) kernel and geometric-mean-comparison kernel – on their performance. Experiments on the NIST 2010 Speaker Recognition Evaluation suggest that GMM-SVM with UP-AVR is superior to speaker comparison and that the GUMI kernel is slightly better than the KL kernel in speaker comparison.

Index Terms: speaker verification, GMM-SVM, speaker comparison, NIST SRE, utterance partitioning, data imbalance.

1. Introduction

In GMM-SVM [1], the variable-length utterance of a claimant is converted to a fixed-length supervector by stacking the mean vectors of an MAP-adapted Gaussian mixture model (GMM) [2] of the claimant. The supervector is then presented to the target-speaker's support vector machine (SVM) for scoring. The advantage of scoring by SVMs 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].

An unaddressed 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 [4].

We have recently proposed a method called utterance partitioning with acoustic vector resampling (UP-AVR) [4] to address this data imbalance problem. The idea is to create a number of target-speaker's supervectors by partitioning the sequence of acoustic vectors in the enrollment utterance into a

number of segments, with each segment producing one GMM-supervector. To increase the number of supervectors without reducing the segment length (which may compromise their representation power), each segment is constructed by randomly selecting the acoustic vectors from the full-length acoustic vector sequence. This resampling procedure is repeated several times to produce a desirable number of GMM-supervectors. It was demonstrated that this method is effective in overcoming the data imbalance problem and helping the SVM learning algorithm to find a better decision boundary.

Recently, Campbell *et al.* [5] proposed a method called speaker comparison. In this method, speaker scores are formulated as a kernel function of the target-speaker's supervector and the claimant's supervectors. Apparently, this method does not suffer from the data imbalance problem because no training is required. However, as this method does not consider the background speakers in the scoring function, it relies on Z-norm [6] to incorporate background information. All of the Z-norm speakers, however, will be of equal importance, because Z-norm uses the mean score of Z-norm speakers for normalization. On the other hand, in GMM-SVM, information of background speakers is embedded in the target-speakers' SVMs. Instead of assigning equal weights to all background speakers, the SVM training algorithm selects some representative background speakers (corresponding to the support vectors) for each target speaker and assigns optimal weights (corresponding to the Lagrange multipliers) to these speakers. While this seems to be a better way of using the background information than the Z-norm, SVM training suffers from the data imbalance problem. By using UP-AVR to create more speaker-class supervectors, we can make the best use of background information without being hindered by the data imbalance problem.

This paper attempts to answer the following question: "will UP-AVR bring sufficient benefit to SVM training so that the resulting GMM-SVM system can outperform the one that uses speaker comparison only?" To this end, we compared GMM-SVM and speaker comparison under three different kernels: KL divergence kernel [1], GMM-UBM mean interval (GUMI) kernel [7], and geometric-mean-comparison kernel [8]. Our key finding is that GMM-SVM with UP-AVR performs better than speaker comparison. It was also found that under the speaker comparison framework, the GUMI kernel performs better.

2. GMM-SVM with UP-AVR

The idea of GMM-SVM [1] is to make the best use of 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

This work was in part supported by The Hong Polytechnic University (4-ZZ7W) and RGC of the Hong Kong SAR (PolyU 5264/09E).

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

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

where $\alpha_0^{(s)}$ and $\alpha_i^{(s)}$'s are the Lagrange multiplier corresponding to the target speaker¹ and background speakers, respectively; $\mathcal{S}^{(b)}$ is a set containing the indexes of the support vectors in the background-speaker set; $\text{utt}^{(b_i)}$ is the utterance of the i -th background speaker; and $d^{(s)}$ is a bias term. 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 first kernel investigated in this work is the KL divergence kernel [1]:

$$K_{\text{KL}}(\text{utt}^{(c)}, \text{utt}^{(s)}) = \sum_{j=1}^M \left(\sqrt{\lambda_j} \Sigma_j^{-\frac{1}{2}} \mu_j^{(c)} \right)^T \left(\sqrt{\lambda_j} \Sigma_j^{-\frac{1}{2}} \mu_j^{(s)} \right) \quad (2)$$

where λ_j and Σ_j are the mixture weights and covariances of the UBM, respectively, and $\mu_j^{(s)}$ and $\mu_j^{(c)}$ are the j -th mean vector of the GMM belonging to speaker s and claimant c , respectively. The second kernel is the geometric-mean-comparison kernel [8]:

$$C_{\text{GM}}(\text{utt}^{(c)}, \text{utt}^{(s)}) = \sum_{j=1}^M \left(\sqrt{\lambda_j^{(c)}} \Sigma_j^{-\frac{1}{2}} \mu_j^{(c)} \right)^T \left(\sqrt{\lambda_j^{(s)}} \Sigma_j^{-\frac{1}{2}} \mu_j^{(s)} \right) \quad (3)$$

where $\lambda_j^{(s)}$ and $\lambda_j^{(c)}$ are the j -th mixture weight. The KL and C_{GM} kernels are different in that the mixture weights of the former are speaker-independent whereas the mixture weights of the latter are speaker-dependent. The third kernel is the GMM-UBM mean interval (GUMI) kernel [7]:

$$K_{\text{GUMI}}(\text{utt}^{(c)}, \text{utt}^{(s)}) = \sum_{j=1}^M \left[\left(\frac{\Sigma_j^{(c)} + \Sigma_j^{(u)}}{2} \right)^{-\frac{1}{2}} \left(\mu_j^{(c)} - \mu_j^{(u)} \right) \right]^T \left[\left(\frac{\Sigma_j^{(s)} + \Sigma_j^{(u)}}{2} \right)^{-\frac{1}{2}} \left(\mu_j^{(s)} - \mu_j^{(u)} \right) \right] \quad (4)$$

where u , s and c represent the UBM, speaker, and claimant, respectively.

In typical GMM-SVM setting, there is only one speaker-class's supervector for training. Utterance partitioning with acoustic vector resampling (UP-AVR) [4] was proposed to increase the influence of speaker-class data on the decision plane. This approach firstly partitions an enrollment utterance into a number of 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. To produce a sufficient number of sub-utterances without compromising their representation power, UP-AVR uses the notion of random resampling in bootstrapping. The idea is based on the fact that 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

¹We assume one enrollment utterance per target speaker.

precisely, if this process is repeated R times, we obtain RN sub-utterances from a single enrollment utterance. In this work, $R = 1$ and $N = 4$. The effect of varying R and N on speaker verification performance can be found in [4].

3. Speaker Comparison with Inner Product Discriminant Functions

Unlike GMM-SVM, speaker comparison [5] does not require the training of an SVM for each target speaker, thereby avoiding the data imbalance problem. The method computes the score of a test utterance (produced by a claimant) by evaluating the inner product between the claimant's supervector and target-speaker's supervector. The score is then compared with a threshold:

$$S_{\text{SC}}(\text{utt}^{(c)}, \text{utt}^{(s)}) = K(\text{utt}^{(c)}, \text{utt}^{(s)}) \leq \theta, \quad (5)$$

where the kernel K can be any valid kernel functions such as Eqs. 2–4. A comparison between Eq. 5 and Eq. 1 reveals that speaker comparison is a special case of GMM-SVM scoring in which no background information is used. In particular, because speaker comparison does not compute the similarity between the claimant's utterance and background utterances, score normalization (such as Z-norm) is very important for the success of this method. Our experimental results in Section 5 also suggest that this is the case.

4. Experiments

4.1. Speech Data and Acoustic Features

The NIST 2010 Speaker Recognition Evaluation (SRE)² was used for performance evaluation. This paper focuses on the interview and microphone speech of the core task, i.e., Common Conditions 1, 2, 4, 7 and 9. NIST 2005–2008 SREs were used as development data (NAP, UBM, Z-norm and T-norm). Only male speakers in these corpora were used.

The experiments involve three types of speaker verification methodologies: GMM-SVM [1], speaker comparison with inner product discriminant functions [5], and joint factor analysis (JFA) [9]. Silence regions of the utterances in these corpora were removed by a VAD [10]. For GMM-SVM and speaker comparison, 12 MFCCs plus their first derivative were extracted from the speech regions of the utterances, leading to 24-dim acoustic vectors. Cepstral mean normalization was then applied to the MFCCs, followed by feature warping [11]. For the JFA system, 19 MFCCs plus their 1st- and 2nd- derivatives were extracted from the speech regions of each utterance, leading to 60-dim acoustic vectors. Cepstral mean normalization and feature warping with a window of 3 seconds were applied to the acoustic vectors.

4.2. Kernel Scoring and Normalization

The GMM-SVM and speaker comparison systems use three different kernels, including the KL divergence kernel K_{KL} , GMM-UBM mean interval kernel K_{GUMI} , and the geometric-mean-comparison kernel C_{GM} . The GMM-supervectors (speaker models) for these kernels were adapted from a 512-Gaussian UBM created from a subset (totally 5,077 utterances) of microphone speech in NIST05–06.³ Different MAP adaptation [2]

²<http://www.itl.nist.gov/iad/mig/tests/sre>

³Hereafter, all NIST SREs are abbreviated as NIST XX , where XX stands for the year of evaluation.

parameters were used to create the supervectors for different kernels. Specifically, for the KL kernel, only the means were adapted, using a relevance factor of 16. For the GUMI kernel, a relevance factor of 16 was used to adapt the means and variances. For the geometric-mean-comparison kernel, the relevance factors for the means and mixture weights were set to 0.01 and zero, respectively, meaning that the mixture weights were estimated using maximum-likelihood principle with mixture indexes aligned with those of the UBM. We followed the setting of the relevance factors as in [8]. The reason of using such a small relevance factors will be explained in Section 5.2.

Unless stated otherwise, ZT-norm [6] was used for score normalization. For GMM-SVM, 300 T-norm models were created from the microphone utterances in NIST05 and 300 Z-norm utterances were selected from NIST05–06. For JFA, 288 T-norm utterances and 288 Z-norm utterances were selected from the microphone speech in NIST05–08.

4.3. Session and Channel Variability Modelling

For the GMM-SVM and speaker comparison systems, we applied NAP [1] to all GMM-supervectors for channel compensation. We selected 143 male speakers from NIST05–08 for estimating the projection matrix. Each of these speakers has eight or more utterances recorded by different microphones.

The JFA system is based on 1024-Gaussian UBMs. Due to insufficient microphone data, we used both telephone and microphone utterances to estimate the eigenvoice and eigenchannel matrices. Specifically, for the eigenvoice matrices, we used 1,194 speakers from NIST04–08, Switchboard II Phase 2 and Phase 3, and Switchboard Cellular Parts 2 to estimate a telephone eigenvoice matrix with a speaker factor of 230. For the microphone channel, we selected 4,072 utterances from 144 speakers (each with at least 5 utterances) in NIST05–08 to estimate a 120-factor eigenvoice matrix. Then, the telephone and microphone eigenvoice matrices were concatenated to produce a 61400×350 eigenvoice matrix.

To estimate an eigenchannel matrix with 100 channel factors for the telephone channel, we selected 8,795 telephone utterances from 806 speakers in NIST04–08 and Switchboard Cellular Parts 2. We used the same set of data that used for estimating the microphone eigenvoice matrix to estimate a 50-factor microphone eigenchannel matrix. The two channel-dependent eigenchannel matrices were then concatenated to form a 61400×150 eigenchannel matrix. The concatenated eigenvoice and eigenchannel matrices were then used for estimating the speaker residual matrix. A modified version of the BUT JFA Matlab Demo was used for JFA training and scoring.

5. Results and Discussions

5.1. Comparing Kernels in GMM-SVM

Table 1 shows that under the GMM-SVM framework, the performance of the KL kernel is comparable to that of the GUMI kernel and that the C_{GM} kernel performs poorly. One possible reason for the poor performance of the C_{GM} kernel in GMM-SVM is the small relevance factor ($r = 0.01$) in the MAP adaptation. A small relevance factor means that the test- and target-speaker supervectors depend almost exclusively on the test and enrollment utterances, respectively. In other words, the resulting supervectors lose all of the background information contained in the UBM. As GMM-SVM harnesses the background information via the background speakers’ supervectors, a small relevance factor means that these supervectors only represent

some speakers different from the target speaker. As a result, a small number of background speakers – which in our case is only 300 – may not be able to represent the background population. To confirm this conjecture, we have tried increasing the relevance factors for adapting the means from 0.01 to 16 and found that the performance improves when the relevance factor increases. This suggests that to work with GMM-SVM, the C_{GM} kernel requires a larger relevance factor, e.g., $r = 8$.

5.2. Comparing Kernels in Speaker Comparison

Table 1 shows that under the speaker comparison framework, the GUMI kernel outperforms the KL kernel and the C_{GM} kernel in terms of EER. As the GUMI kernel is derived from the Bhattacharyya distance, which is a better similarity measure between two probability distributions than the simplified KL divergence, its performance is better.

Unlike the GMM-SVM, we observed that under the speaker comparison framework, the C_{GM} kernel requires a small relevance factor ($r = 0.01$). It is of interest to understand why the same kernel has different requirements under different scoring framework. Consider the case where $r = 16$. With such a large relevance factor, only the Gaussians that are close to the adaptation data will shift towards the adaptation data, the rest will remain unadapted or only slightly shift in position. This means that these unadapted Gaussians will only occupy a small amount or even no adaptation data. However, in C_{GM} , the mixture weights are estimated using the maximum-likelihood principle, meaning that the mixture weights of the unadapted Gaussians will be very small or even zero. This is undesirable because the extremely small mixture weights severely suppress the contribution of the corresponding components in the inner product, causing loss of speaker information. On the other hand, if the relevance factor is small, say $r = 0.01$, almost all Gaussians will shift towards the adaptation data, resulting in a more even sharing of adaptation data among the Gaussians. This is desirable because almost all mixture components will have contribution to the scoring function. Evidences supporting this argument can be found in Fig. 1. It shows that when the relevance factor for the means become large, the number of small mixture weights (many of them are zero) increases.

Because no adaptation is applied to the mixture weights in the KL and GUMI kernels, the over-suppression phenomenon will not occur. So, a larger relevance factor for these two kernels can be used.

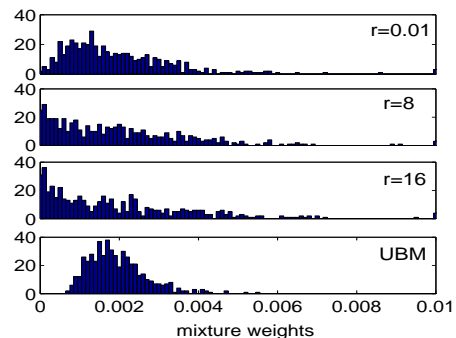


Figure 1: Histograms of mixture weights with increasing relevance factor r for the means in the C_{GM} kernel. The bottom panel shows the histogram of UBM’s mixture weights.

5.3. GMM-SVM vs. Speaker Comparison

Results in Table 1 also suggest that the performance of GMM-SVM with a KL kernel is better than that of the speaker com-

System	EER (%)					minNDCF				
	CC1	CC2	CC4	CC7	CC9	CC1	CC2	CC4	CC7	CC9
(A) SVM K_{KL}	2.82	5.21	3.81	6.70	4.27	0.61	0.62	0.63	0.63	0.19
(B) SVM K_{GUMI}	2.83	5.39	4.06	6.99	4.27	0.57	0.61	0.61	0.67	0.24
(C) SVM C_{GM}	4.85	7.31	5.71	7.82	5.13	0.77	0.83	0.83	0.58	0.26
(D) SVM K_{KL} + UP-AVR	2.22	5.37	3.43	5.59	3.42	0.53	0.75	0.61	0.64	0.21
(E) SVM K_{KL} + UP-AVR (T-norm Only)	2.02	4.91	3.35	5.59	4.27	0.44	0.70	0.58	0.58	0.13
(F) SVM K_{GUMI} + UP-AVR	2.63	5.48	3.51	5.52	4.27	0.52	0.74	0.58	0.69	0.18
(G) SVM K_{GUMI} + UP-AVR (T-norm Only)	2.53	5.22	3.43	5.57	4.27	0.39	0.67	0.54	0.59	0.15
(H) SC K_{KL}	3.94	6.81	3.91	7.26	5.00	0.57	0.69	0.52	0.69	0.36
(I) SC K_{GUMI}	2.93	5.25	4.07	7.04	4.27	0.65	0.65	0.68	0.64	0.26
(J) SC C_{GM}	3.44	5.66	4.37	6.70	4.24	0.54	0.65	0.75	0.77	0.45
(K) SC C_{GM} (T-norm Only)	4.65	9.56	6.53	11.13	7.69	0.50	0.71	0.66	0.80	0.68
(L) JFA	2.72	4.75	3.90	6.14	4.27	0.46	0.53	0.67	0.68	0.22
(E)+(L)	1.62	3.49	2.69	5.52	4.25	0.31	0.44	0.48	0.59	0.19

Table 1: The performance of different systems on NIST 2010 SRE under different common conditions (CC). Results were divided into three groups: (1) SVM – GMM-SVM with different kernels, (2) SC – speaker comparison using different inner-product discriminant kernels, and (3) JFA – joint factor analysis models. The lowest EER and minNDCF across all three groups were displayed in bold. The kernels being compared include the KL-divergence kernel K_{KL} , GMM-UBM mean interval kernel K_{GUMI} , and geometric-mean comparison kernel C_{GM} . Except for Systems E, G and K, ZT-norm was applied in all cases. UP-AVR, the method proposed in this paper, was applied to two of the GMM-SVM systems for comparison. (E)+(L) denotes the linear score fusion of the best GMM-SVM system and the JFA system.

parison with a GUMI kernel and a C_{GM} kernel. Comparing Systems J and K in the Table 1 shows that Z-norm plays an important role in speaker comparison. This is because the scoring function in speaker comparison does not consider the background speakers. Therefore, it is important to harness the impostor information through score normalization methods such as Z-norm. In fact, Z-norm can be considered as a special case of SVM scoring in Eq. 1 where the weights corresponding to all of the background speakers are equal.

5.4. Effect of UP-AVR on GMM-SVM Systems

Table 1 shows that under most of the common conditions, UP-AVR helps improve the performance of GMM-SVM systems for both KL and GUMI kernels. In terms of EER, the performance of UP-AVR in GMM-SVM systems (system E) is consistently better than that of the speaker comparison systems and the JFA system (except for Common Condition 2). This further demonstrates that UP-AVR is effective in solving the imbalance data problem in GMM-SVM systems. In this paper, we only show the result of UP-AVR with 5 GMM-supervectors, which have already been enough to alleviate the imbalance data problem. As demonstrated in our earlier work [4], using a large number of positive supervectors will only result in a large number of zero Lagrange multipliers for speaker class without substantial gain in speaker verification performance.

Interestingly, ZT-norm is inferior to T-norm in GMM-SVM with UP-AVR. However, we found that (result not shown) applying T-norm only to GMM-SVM systems without UP-AVR leads to poor performance. The reason might be that we did not apply UP-AVR to the Z-norm utterances.

The scores of JFA and GMM-SVM with UP-AVR (KL kernel, T-norm only) were fused using a set of linear fusion weights that achieve the best fusion performance (in terms of minimum EER). Table 1 shows that the fusion improves the performance for Common Conditions 1, 2, and 4.

6. Conclusion

This paper has demonstrated that creating a few more speaker-class supervectors for SVM training by using the utterance par-

tioning technique will bring significant benefit to GMM-SVM systems. It was found that the resulting GMM-SVM systems are competitive with and sometimes superior to the state-of-the-art speaker comparison systems.

7. References

- [1] 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.
- [2] 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.
- [3] S. X. Zhang and M. W. Mak, "Optimized Discriminative Kernel for SVM Scoring and Its Application to Speaker Verification," *IEEE Trans. on Neural Networks*, no. 2, pp. 173–185, 2011.
- [4] 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.
- [5] W. M. Campbell, Z. Karam, and D. E. Sturim, "Speaker comparison with inner product discriminant functions," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 207–215.
- [6] 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.
- [7] C. H. You, K. A. Lee, and H. Z. Li, "GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition," *IEEE Trans on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1300–1312, 2010.
- [8] W. M. Campbell and Z. N. Karam, "Simple and efficient speaker comparison using approximate KL divergence," in *Proc. Interspeech 2010*, Japan, 2010.
- [9] 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.
- [10] M. W. Mak and H. B. Yu, "Robust voice activity detection for interview speech in nist speaker recognition evaluation," in *Proc. APSIPA ASC 2010*, Singapore, 2010.
- [11] 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.