



Simple and Efficient Speaker Comparison using Approximate KL Divergence*

W. M. Campbell[†], Z. N. Karam^{†‡}

[†]MIT Lincoln Laboratory, Lexington, MA

[‡]DSPG, Research Laboratory of Electronics at MIT, Cambridge MA

Abstract

We describe a simple, novel, and efficient system for speaker comparison with two main components. First, the system uses a new approximate KL divergence distance extending earlier GMM parameter vector SVM kernels. The approximate distance incorporates data-dependent mixture weights as well as the standard MAP-adapted GMM mean parameters. Second, the system applies a weighted nuisance projection method for channel compensation. A simple eigenvector method of training is presented. The resulting speaker comparison system is straightforward to implement and is computationally simple—only two low-rank matrix multiplies and an inner product are needed for comparison of two GMM parameter vectors. We demonstrate the approach on a NIST 2008 speaker recognition evaluation task. We provide insight into what methods, parameters, and features are critical for good performance.

Index Terms: speaker recognition

1. Introduction

Text-independent speaker comparison is the process of taking two speech utterances and providing a match score or posterior probability of match. Speaker comparison can be considered to be a core building block for building speaker recognition systems. Standard approaches to comparison include training and testing using a classifier or building speaker utterance kernels, e.g. [1, 2].

Speaker comparison can be implemented using many different classifiers. We focus on approaches using a GMM universal background model (GMM UBM). Speaker comparison is accomplished using SVM kernel techniques [2]. In this structure, a GMM UBM is adapted per utterance and the resulting models are compared using an approximate KL divergence. This framework is simple and intuitive for speaker recognition since utterances are represented using GMM parameter vectors and speaker comparison is a simple inner product.

Significant improvements in error rates for speaker comparison can be obtained by using data-driven subspace models for channel and speaker representation. Two significant approaches are nuisance attribute projection (NAP) and joint factor analysis (JFA). NAP [2] uses a fixed orthogonal projection to remove nuisance directions from the GMM parameter vector. Typically, this nuisance is modeled as session variation. JFA [3] models both the speaker and session variation with subspaces. Factors (coordinates) for the subspaces are derived using a MAP criterion with a prior on the factors.

Combining comparison methods with subspace methods was studied extensively in the inner product discriminant func-

tion (IPDF) framework [4]. IPDFs considered numerous combinations of classifiers and compensation methods and found two key aspects of good performance. First, classification methods incorporating *both* speaker-dependent mean and mixture-weight parameters gave significant improvement over mean-only systems. Second, subspace channel compensation provided the bulk of system performance improvements.

In this paper, we present an approximate KL divergence kernel combined with weighted NAP (WNAP) [5] that implements the key insights from the IPDF framework. Our strategy is to focus on an easy to implement system that is efficient and achieves state-of-the-performance. An added bonus of our result is that our resulting method is an SVM kernel and can be used in future work for other speaker recognition tasks.

In this paper, we first cover the top-level speaker comparison framework in Section 2, then we present an approximate KL-divergence method in Section 3. Section 4 discusses WNAP and the corresponding training criterion. Section 5 presents algorithms for speaker comparison method. Finally, experiments in Section 6 demonstrate the effectiveness of the method and provide insight into key methods for achieving good performance.

2. GMM Parameter Vectors

A standard distribution used for text-independent speaker recognition is the Gaussian mixture model [6],

$$g(\mathbf{x}) = \sum_{i=1}^N \lambda_i \mathcal{N}(\mathbf{x} | \mathbf{m}_i, \Sigma_i). \quad (1)$$

Feature vectors are typically cepstral coefficients with associated smoothed first- and second-order derivatives.

A sequence of feature vectors, $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{N_x})$, from a speaker is mapped to a GMM by adapting a GMM universal background model (UBM). We will assume only the mixture weights, λ_i , and means, \mathbf{m}_i , in (1) are adapted. Adaptation of the means is performed with standard relevance MAP [6]. We estimate the mixture weights using the standard ML estimate. The adaptation yields new parameters which we stack into a parameter vector, \mathbf{p}_x , where

$$\mathbf{p}_x = [\boldsymbol{\lambda}_x^t \quad \mathbf{m}_x^t]^t \quad (2)$$

$$= [\lambda_{x,1} \quad \dots \quad \lambda_{x,N} \quad \mathbf{m}_{x,1}^t \quad \dots \quad \mathbf{m}_{x,N}^t]^t. \quad (3)$$

Speaker comparison is the process of comparing two sequences of feature vectors, \mathbf{X} and \mathbf{Y} . Rather than compare these directly, we compare the corresponding parameter vectors, \mathbf{p}_x and \mathbf{p}_y , obtained from separately adapting the GMM UBM to \mathbf{X} and \mathbf{Y} . The goal is to provide a comparison function $C(\mathbf{p}_x, \mathbf{p}_y)$ that produces a value reflecting the similarity of the speakers represented by the two parameter vectors.

*This work was sponsored by the Federal Bureau of Investigation under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

3. Approximate KL Divergence

An obvious strategy for comparing the GMM parameter vectors is to use the KL divergence between the distributions

$$D(g_x \| g_y) = \int_{R^n} g_x(\mathbf{x}) \log \left(\frac{g_x(\mathbf{x})}{g_y(\mathbf{x})} \right) d\mathbf{x}. \quad (4)$$

Using the KL divergence directly is difficult because it cannot be computed in closed form. Therefore, an approximate KL divergence has been used successfully in speaker recognition [2].

An approximation based on the log-sum inequality is applied to (4) to split out individual mixtures to obtain

$$D(g_x \| g_y) \leq \sum_{i=1}^N \lambda_{x,i} D(\mathcal{N}(\cdot; \mathbf{m}_{x,i}, \Sigma_i) \| \mathcal{N}(\cdot; \mathbf{m}_{y,i}, \Sigma_i)). \quad (5)$$

Here, Σ_i is from the UBM. Note that we have dropped the term $D(\lambda_x \| \lambda_y)$, since we are finding an upper bound and the KL divergence is always greater than zero.

By symmetrizing (5) and substituting in the KL divergence between two Gaussian distributions, we obtain a distance which upper bounds the symmetric KL divergence, $d_s(\mathbf{p}_x, \mathbf{p}_y)$,

$$\sum_{i=1}^N (0.5\lambda_{x,i} + 0.5\lambda_{y,i}) (\mathbf{m}_{x,i} - \mathbf{m}_{y,i})^t \Sigma_i^{-1} (\mathbf{m}_{x,i} - \mathbf{m}_{y,i}). \quad (6)$$

A corresponding inner product to this distance is

$$C_{\text{KL}}(\mathbf{p}_x, \mathbf{p}_y) = \sum_{i=1}^N (0.5\lambda_{x,i} + 0.5\lambda_{y,i}) \mathbf{m}_{x,i}^t \Sigma_i^{-1} \mathbf{m}_{y,i}. \quad (7)$$

Note that (7) can also be expressed more compactly as

$$C_{\text{KL}}(\mathbf{p}_x, \mathbf{p}_y) = \mathbf{m}_x^t ((0.5\lambda_x + 0.5\lambda_y) \otimes I_n) \Sigma^{-1} \mathbf{m}_y \quad (8)$$

where Σ is the block matrix with the Σ_i from the UBM on the diagonal, n is the feature vector dimension, and \otimes is the Kronecker product. Note that shifting the means by the UBM will not affect the distance in (6), so we can replace means in (8) by the UBM centered means.

The comparison function, C_{KL} , does not correspond to an inner product in the Mercer sense. That is, we cannot separate C_{KL} into an inner product of the form $\mathbf{b}(\mathbf{p}_x)^t \mathbf{b}(\mathbf{p}_y)$ where $\mathbf{b}(\cdot)$ is some mapping function. A simple solution to this problem is to replace the arithmetic mean between mixture weights in (8) with a geometric mean; we obtain

$$C_{\text{GM}}(\mathbf{p}_x, \mathbf{p}_y) = \mathbf{m}_x^t (\lambda_x^{1/2} \otimes I_n) \Sigma^{-1} (\lambda_y^{1/2} \otimes I_n) \mathbf{m}_y \quad (9)$$

where Σ is the block diagonal of the UBM covariances. In experiments, we have found (9) to be a good approximation of (8).

We mention that the corresponding SVM expansion to the kernel (9) is

$$\mathbf{b}(\mathbf{p}_x) = (\lambda_x^{1/2} \otimes I_n) \Sigma^{-1/2} \mathbf{m}_x. \quad (10)$$

4. W NAP

Before defining W NAP, we introduce some notation. We define an orthogonal projection with respect to a metric, $P_{U,D}$, where D and U are full rank matrices as

$$P_{U,D} = U(U^t D^2 U)^{-1} U^t D^2 \quad (11)$$

where DU is a linearly independent set, and the metric is

$$\|x - y\|_D = \|Dx - Dy\|_2. \quad (12)$$

The process of projection, e.g. $y = P_{U,D}b$, is equivalent to solving the least-squares problem,

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|Ux - b\|_D \quad (13)$$

and letting $y = U\hat{x}$. For convenience, we also define the projection onto the orthogonal complement of U , U^\perp , as $Q_{U,D} = P_{U^\perp,D} = I - P_{U,D}$. In practice, the projection (11) is implemented as a matrix multiply by orthonormalizing the basis U for the subspace with respect to the appropriate metric.

For W NAP, we use a general projection onto U^\perp of the form Q_{U,D_i} . The use of this projection is to reduce nuisances present in the expansion proposed in Section 2. The main assumption is that the nuisance is confined to a ‘‘small’’ dimensional subspace of the expansion space.

For the W NAP training set, we assume that for every speaker (in general, every class), that we can estimate a ‘‘low noise’’ vector $\bar{\mathbf{x}}$ from which deltas can be calculated. In practice, this smoothed vector is formed by adapting a model from the data pooled across multiple utterances from the same speaker. We then base our criterion on approximating these deltas.

More specifically, suppose we have a training set, $\{\mathbf{z}_{s,i}\}$ labeled by speaker, s , and instance, i . For each s , we have a smoothed vector, $\bar{\mathbf{z}}_s$. For W NAP training, we use the following optimization problem,

$$\min_U \sum_s \sum_i W_{s,i} \|P_{U,D_s,i} \delta_{s,i} - \delta_{s,i}\|_{D_{s,i}}^2 \quad (14)$$

where $\delta_{s,i} = \mathbf{z}_{s,i} - \bar{\mathbf{z}}_s$. The W NAP training criterion (14) incorporates the goals of using a variable metric and an utterance dependent weighting, $W_{s,i}$, see [5]. The training criterion attempts to find a subspace U that best approximates the nuisance $\delta_{s,i}$ as in prior work [2].

For the purposes of this work, we assume that $D_{s,i} = D$ is a constant. Prior work has shown that this is a good compromise in performance and computational efficiency [5]. In the case of constant D , the W NAP criterion can be shown to be equivalent to the following problem. First, we incorporate the W_i into the δ_i by letting,

$$\hat{\delta}_i = \sqrt{W_i} \delta_i. \quad (15)$$

Second, we find the correlation matrix,

$$R = \sum_{i=1}^N \hat{\delta}_i \hat{\delta}_i^t. \quad (16)$$

Then, the criterion (14) can be expressed as,

$$\max_{\hat{U}, \hat{U}^t \hat{U} = I} \operatorname{tr} [\hat{U}^t \hat{R} \hat{U}] \quad (17)$$

In the equation, U is the desired nuisance subspace, $\hat{U} = DU$, and $\hat{R} = DRD$. This problem can be solved using an eigenvector method that will be presented in Section 5. Intuitively, the problem (17) finds a low rank approximation, U , that best approximates the nuisance subspace.

5. Algorithms

Our method for speaker comparison can be split into two components—training the nuisance subspace and performing speaker comparison scoring. Both algorithms are straightforward to implement with matrix tools such as Matlab.

Training the nuisance subspace is shown in Algorithm 1. For the training set, we first note that only the utterance MAP-adapted means for the vectors \mathbf{m}_i are used. A typical data set for training the nuisance subspace would have several sessions per speaker—typically 8 or more. A second comment on Algorithm 1 is that the metric, D , used for training the subspace is not utterance dependent. In practice, this has not impacted performance. Third, we mention that one good choice for W_i is the number of speech frames detected by speech activity detection. Fourth, we mention that in the algorithm, kernel PCA can be used as an alternative to the direct parameter expansion [7].

In Algorithm 2, we show compensation using WNAp and speaker comparison scoring using C_{GM} . Note that the matrix D is the same as in Algorithm 1. We also mention that the relevance factor for MAP adaptation can be tuned. Typically, we use a relevance factor of 0.01.

6. Experiments

6.1. Setup

Experiments were performed on the NIST 2008 speaker recognition evaluation (SRE) data set. Enrollment/verification methodology and the evaluation criterion, equal error rate (EER) and minDCF, were based on the NIST SRE evaluation plan [8]. The main focus of our effort was the one conversation

Algorithm 1 WNAp subspace training algorithm for a fixed metric, $D = (\lambda_{UBM}^{1/2} \otimes I_n) \Sigma^{-1/2}$

Input: Mean parameter vectors $\{\mathbf{m}_i\}$, weights $\{W_i\}$, with speaker labels $\{l_i\}$, and the desired corank
Output: Nuisance subspace, U
for all s in unique speakers in $\{l_i\}$ **do**
 Find $\bar{\mathbf{m}}_s$
 for all j in $\{j | l_j == s\}$ **do**
 Let $\delta_j = \mathbf{m}_j - \bar{\mathbf{m}}_s$
 end for
end for
 $R = 0$
for $i = 1$ to N **do**
 $R = R + W_i \delta_j \delta_j^t$
end for
 $\hat{R} = DRD$
 $\hat{U} = \text{eigs}(\hat{R}, \text{corank})$ % eigs produces the eigenvectors of the largest magnitude eigenvalues
 $U = D^{-1} \hat{U}$

Algorithm 2 Compensation and Scoring with the C_{GM} kernel

Input: Two sequences of feature vectors, \mathbf{X}_1 and \mathbf{X}_2
Output: Comparison score, s
for $i = 1$ to 2 **do**
 \mathbf{p}_i = parameters of MAP adapted UBM to \mathbf{X}_i
 $\mathbf{n}_i = UU^t D^2 \mathbf{m}_i$.
 Update $\mathbf{m}_i \leftarrow \mathbf{m}_i - \mathbf{n}_i$
 $D_i = (\lambda_i^{1/2} \otimes I_n) \Sigma^{-1/2}$
end for
 $s = \mathbf{m}_1^t D_1 D_2 \mathbf{m}_2$

enroll, one conversation verification task for telephone channel speech. T-Norm models and Z-Norm speech utterances were drawn from the NIST 2004 SRE corpus. Results were obtained for both the English only (Eng, pool 7) and for all different number trials (All, pool 6) which includes speakers that enroll/verify in different languages.

Feature extraction was performed using HTK [9] with 20 MFCC coefficients, deltas, and acceleration coefficients for a total of 60 features. Speech activity detection (SAD) was performed using a cascade of two systems. First, a GMM speech/non-speech detector was applied. Then, these SAD marks were post-processed with an energy-based detector. Features from non-speech frames were eliminated and then feature warping [10] was applied to all of the resulting features with a 3 second window.

A GMM UBM with 512 mixture components was trained using data from NIST SRE 2004 and from Switchboard corpora. A nuisance subspace was trained using the speakers from Switchboard 2 and NIST 2004 SRE corpora using Algorithm 1. The dimension of the nuisance subspace, U , was fixed at 64.

A few aspects of the front-end were critical for the best performance. First, the full bandwidth MFCC analysis, 0 – 4 kHz, performed the best. In our experiments, we found that WNAp could take advantage of the additional bandwidth for speaker comparison. Second, our cascaded SAD is fairly aggressive. We found that low-level speech was not helpful in discrimination and could contain cross-talk. Finally, feature warping was a slight gain over feature 0 – 1 mean and variance normalization.

For our SVM system, we used both the mean-only KL kernel, K_{KL} , described in [2] and the new C_{GM} kernel from (9). The SVM background was constructed from Fisher data. T-Norm and Z-Norm were performed in the same manner as the C_{GM} system. A relevance factor of 4 was used for the K_{KL} kernel to match prior work and for best performance. For the C_{GM} kernel, a relevance factor of 0.01 was used in both the kernel-only and SVM experiments.

6.2. Results

The first two lines of Table 1 show baseline systems and their compute time from benchmarks. Computation is not shown for fusion system (MIT LL and MFCC+LPCC) because we are focusing on single system performance. The next three lines of the table contrast the new C_{GM} kernel with the K_{KL} kernel from earlier work. In the table, we see that the new C_{GM} kernel (shown with ZT-norm) outperforms the K_{KL} kernel in all tasks. Note the K_{KL} kernel performs better with a larger relevance factor. Basically, the new C_{GM} incorporates the utterance mixture weights as a way of discounting uncertain mixture components in the inner product; thus, C_{GM} allows a lower relevance factor to be used. The prior kernel, K_{KL} , is more sensitive to a “noisy” model that has mixture components that are uncertain and requires a higher relevance factor.

The next set of experiments were performed using the kernels in an SVM configuration [2]. The experiments demonstrate that the SVM training adds little to the performance of the speaker comparison system. This result might be expected, since having one training vector for the target speaker gives no information about the intra-speaker variation and relies upon non-targets to set the discrimination boundary.

We also performed simple calibration and linear fusion experiments with our system. First, we experimented with different score normalizations, Z-Norm, T-Norm, etc. Second, we demonstrated the effect of not using English/non-English calibration (w/o Calibration) [11]. Third, we fused the MFCC

Table 1: A comparison of different systems on the NIST SRE 2008, one conversation telephone train and test subset (pool 6 and 7). Compute time is normalized to a JFA baseline and includes compensation and inner product only. Best performing systems are shown in bold for reference.

System	EER All (%)	minDCF All ($\times 100$)	EER Eng (%)	minDCF Eng ($\times 100$)	Compute time
BUT MFCC 20 System [11]	5.71	2.95	2.85	1.40	1.00
MIT LL Fused System [12]	7.00	3.60	3.30	1.60	-
$K_{KL}, rf = 0.01$	7.22	3.39	4.40	2.04	0.08
$K_{KL}, rf = 4$	6.11	3.04	3.34	1.68	0.08
C_{GM}	5.86	2.89	3.09	1.57	0.08
SVM K_{KL}	6.51	3.01	3.51	1.65	0.86
SVM C_{GM}	6.31	2.95	3.64	1.57	0.86
C_{GM} w/o ZT-Norm	6.83	3.58	3.75	1.98	0.08
C_{GM} Z-Norm Only	6.39	3.12	3.33	1.64	0.08
C_{GM} T-Norm Only	6.72	3.31	3.63	1.70	0.08
C_{GM} w/o Calibration	6.70	3.68	3.09	1.57	0.08
C_{GM} LPCC	6.18	2.85	2.99	1.59	0.08
C_{GM} Fuse LPCC+MFCC	5.27	2.58	2.91	1.36	-

system score linearly with an LPCC-based system. The LPCC front end was a minor change of the HTK configuration—18 cepstral coefficients with energy were used along with deltas and acceleration for a total of 57 features. The resulting system also performed well and fused with our base MFCC system to achieve substantially better performance. Fusion of multiple feature types with the same system has improved performance in many systems [11, 13].

6.3. Analysis

We note that our best performing system, C_{GM} with ZT-Norm, performs well in comparison to other systems in the literature [11] and [12]. A key point of our current work is that our computation and implementation is simplified with respect to previous methods.

Our new C_{GM} system with WNAP reduces complexity over older systems with similar performance. First, we have shown that the SVM training in [12] is not necessary for the speaker comparison task. Second, our system does not require the use of *joint* factor analysis (JFA) [3]. JFA requires considerably more resources in both corpora and computation. For the speaker subspace in JFA, a large corpora is needed to model interspeaker variation. For computation, the JFA system requires the solution of both speaker and channel factors. Compensation and scoring with JFA can be an order of magnitude slower [4] in our benchmarks. In both the SVM and JFA case, further research is needed to understand performance in tasks where more (or less) speaker data is available.

7. Conclusions

A new kernel for speaker comparison based upon an approximate KL divergence was presented. We showed that subspace-based channel compensation could be trained and implemented with a simple algorithm, WNAP. An analysis of various configurations of the system demonstrated that simple approximate KL scoring and WNAP produced excellent performance in comparison to SVM and JFA systems. Several methods for achieving state-of-the-art performance were presented.

8. References

[1] W. M. Campbell, “Generalized linear discriminant sequence kernels for speaker recognition,” in *ICASSP*, 2002, pp. 161–164.

[2] 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 *ICASSP*, 2006, pp. 197–1100.

[3] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, “A study of inter-speaker variability in speaker verification,” *IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[4] W. M. Campbell, Z. Karam, and D. E. Sturim, “Speaker comparison with inner product discriminant functions,” in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 207–215.

[5] W. M. Campbell, “Weighted nuisance attribute projection,” in *submitted to Proc. Odyssey 2010: The Speaker and Language Recognition Workshop*, 2010.

[6] Douglas A. Reynolds, T. F. Quatieri, and R. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[7] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller, “Kernel principal component analysis,” in *Advances in Kernel Methods*, Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, Eds., pp. 327–352. MIT Press, Cambridge, Massachusetts, 1999.

[8] M. A. Przybocki, A. F. Martin, and A. N. Le, “NIST speaker recognition evaluations utilizing the Mixer corpora—2004,2005,2006,” *IEEE Trans. on Speech, Audio, Lang.*, vol. 15, no. 7, pp. 1951–1959, 2007.

[9] J. Odell, D. Ollason, P. Woodland, S. Young, and J. Jansen, *The HTK Book for HTK V2.0*, Cambridge University Press, Cambridge, UK, 1995.

[10] J. Pelecanos and S. Sridharan, “Feature warping for robust speaker verification,” in *Proc. of Speaker Odyssey Workshop*, 2001, pp. 213–218.

[11] L. Burget, V. Hubeika, O. Glembek, M. Karafiat, M. Kockmann, P. Matejka, Petr Schwarz, and J. Cernocky, “BUT system for NIST 2008 speaker recognition evaluation,” in *Proc. Interspeech*, 2009, pp. 2335–2338.

[12] D. Sturim, W. M. Campbell, Z. Karam, D. A. Reynolds, and F. Richardson, “The MIT Lincoln Laboratory 2008 speaker recognition system,” in *Proc. Interspeech*, 2009, pp. 2359–2362.

[13] W. M. Campbell, D. E. Sturim, W. Shen, D. A. Reynolds, and J. Navrátil, “The MIT-LL/IBM 2006 speaker recognition system: High-performance reduced-complexity recognition,” in *ICASSP*, 2007, pp. IV–217–IV–220.