Support Vector Machines Using GMM Supervectors for Speaker Verification

W. M. Campbell, *Member, IEEE*, D. E. Sturim, *Member, IEEE*, and D. A. Reynolds, *Senior Member, IEEE*

***Abstract*—Gaussian mixture models (GMMs) have proven extremely successful for text-independent speaker recognition. The standard training method for GMM models is to use MAP adaptation of the means of the mixture components based on speech from a target speaker. Recent methods in compensation for speaker and channel variability have proposed the idea of stacking the means of the GMM model to form a GMM mean supervector. We examine the idea of using the GMM supervector in a support vector machine (SVM) classifier. We propose two new SVM kernels based on distance metrics between GMM models. We show that these SVM kernels produce excellent classification accuracy in a NIST speaker recognition evaluation task.**

***Index Terms*—Gaussian mixture models (GMMs), speaker recognition, support vector machines (SVMs).**

I. Introduction

We consider the problem of text-independent speaker verification. That is, given a test utterance, a claim of identity, and the corresponding speaker model, determine if the claim is true or false. The standard approach to this problem is to model the speaker using an adapted Gaussian mixture model (GMM) [1].

An exciting area of recent work in GMM speaker recognition is the use of latent factor analysis to compensate for speaker and channel variability [2]. These methods work by modeling the MAP adapted means of a GMM using latent factors to describe variation. A key method in this approach is to use a GMM supervector consisting of the stacked means of the mixture components. This GMM supervector can be used to characterize the speaker and channel using eigenvoices and eigenchannels methods, respectively [3].

Support vector machines (SVMs) have proven to be a new effective method for speaker recognition [4], [5]. SVMs are a natural solution to the problem, since speaker verification is fundamentally a two-class problem. We want to decide between the hypothesis that the speech is produced from the speaker or the hypothesis that the speech is produced from someone else in the population. SVMs perform a nonlinear mapping from an input space to an SVM feature space. Linear classification techniques are then applied in this potentially high-dimensional space. The main design component in an SVM is the kernel, which is an inner product in the SVM feature space. Since inner products induce distance metrics and vice versa, the basic goal in SVM kernel design is to find an appropriate metric in the SVM feature space relevant to the classification problem.

In this letter, we combine the recent results in SVM methods with the GMM supervector concept. We show two natural methods for finding distances between GMM supervectors. One method is based upon an approximation to KL divergence between two GMM models, and the other method is based upon an L^2 function space inner product. Both of these distances satisfy the Mercer condition typically required in SVM optimization.

The outline of the letter is as follows. In Section II, we describe the basic framework for SVMs. In Section III, we outline the GMM supervector expansion. Sections IV and V describe two kernels for SVM speaker verification. Finally, in Section VI, we demonstrate the potential of the approach by applying it to a NIST speaker recognition evaluation 2005 task and compare it to a standard GMM approach.

II. Support Vector Machines

An SVM [6] is a two-class classifier constructed from sums of a kernel function $K(\cdot,\cdot)$

$$f(\mathbf{x}) = \sum_{i=1}^{L} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d \tag{1}$$

where the t_i are the ideal outputs, d is a learned constant, $\sum_{i=1}^{L} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization process [7]. The ideal outputs are either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1, respectively. For classification, a class decision is based upon whether the value $f(\mathbf{x})$ is above or below a threshold.

The kernel $K(\cdot,\cdot)$ is constrained to have certain properties (the Mercer condition), so that $K(\cdot,\cdot)$ can be expressed as

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{b}(\mathbf{x})^t \mathbf{b}(\mathbf{y}) \tag{2}$$

where $\mathbf{b}(\mathbf{x})$ is a mapping from the input space (where $\mathbf{x}$ lives) to a possibly infinite dimensional expansion space. The Mercer condition ensures that the margin concept is appropriate, and the optimization of the SVM is well defined.

The optimization condition relies upon a maximum margin concept. For a separable data set, the system places a hyperplane in a high-dimensional space so that the hyperplane has maximum margin. The data points from the training set lying on the boundaries are the support vectors in (1). The focus, then, of the SVM training process is to model the boundary between classes.

Manuscript received September 19, 2005; revised November 24, 2005. This work was supported by the Department of Defense 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. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dominic K. C. Ho.

The authors are with the MIT Lincoln Laboratory, Lexington, MA 02420 USA (e-mail: wcampbell@ll.mit.edu).

Digital Object Identifier 10.1109/LSP.2006.870086

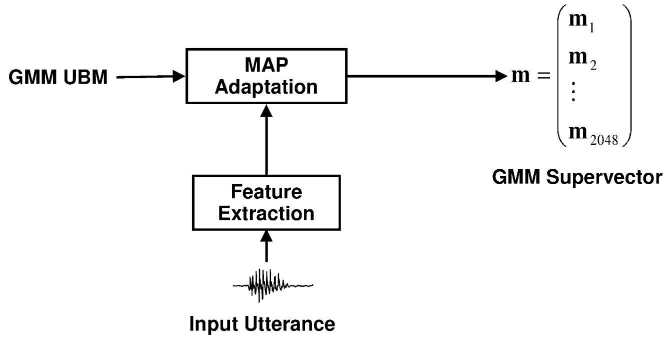

Fig. 1. GMM supervector concept.

III. GMM Supervectors

Suppose we have a Gaussian mixture model universal background model (GMM UBM)

$$g(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(\mathbf{x}; \mathbf{m}_i, \mathbf{\Sigma}_i) \tag{3}$$

where λ_i are the mixture weights, $\mathcal{N}()$ is a Gaussian, and $\mathbf{m}_i$ and $\mathbf{\Sigma}_i$ are the mean and covariance of the Gaussians, respectively. We assume diagonal covariances $\mathbf{\Sigma}$.

Given a speaker utterance, GMM UBM training is performed by MAP adaptation [1] of the means $\mathbf{m}_i$. From this adapted model, we form a GMM supervector. The process is shown in Fig. 1.

The GMM supervector can be thought of as a mapping between an utterance and a high-dimensional vector. This concept fits well with the idea of an SVM sequence kernel [4]. The basic idea of a sequence kernel is to compare two speech utterances utt_a and utt_b directly with a kernel $K(utt_a, utt_b)$. The kernel can be written as $K(utt_a, utt_b) = \mathbf{b}(utt_a)^t \mathbf{b}(utt_b)$ because of the Mercer condition. The GMM supervector mapping is then part of the mapping of utt_a to $\mathbf{b}(utt_a)$. For the simple case of a linear kernel, mapping utt_a to the GMM supervector $\mathbf{m}^a$ is $\mathbf{b}(utt_a)$.

IV. GMM Supervector Linear Kernel

Suppose we have two utterances utt_a and utt_b. We train GMMs g_a and g_b as in (3), on the two utterances, respectively, using MAP adaptation. A natural distance between the two utterances is the KL divergence

$$D(g_a \| g_b) = \int_{R^n} g_a(x) \log\left(\frac{g_a(x)}{g_b(x)}\right) dx. \tag{4}$$

Unfortunately, the KL divergence does not satisfy the Mercer condition, so using it in an SVM is difficult (although possible—see [8]).

Instead of using the divergence directly, we consider an approximation. The idea is to bound the divergence using the log-sum inequality [9]

$$D(g_a \| g_b) \le \sum_{i=1}^{N} \lambda_i D\left(\mathcal{N}\left(\cdot; \mathbf{m}_i^a, \mathbf{\Sigma}_i\right) \| \mathcal{N}\left(\cdot; \mathbf{m}_i^b, \mathbf{\Sigma}_i\right)\right) \tag{5}$$

where we have represented the adapted supervector of means by $\mathbf{m}^a$ and $\mathbf{m}^b$. Assuming diagonal covariances, the approximation in (5) can be calculated in closed form as

$$d(\mathbf{m}^a, \mathbf{m}^b) = \frac{1}{2}\sum_{i=1}^{N} \lambda_i \left(\mathbf{m}_i^a - \mathbf{m}_i^b\right)^t \mathbf{\Sigma}_i^{-1} \left(\mathbf{m}_i^a - \mathbf{m}_i^b\right). \tag{6}$$

The final inequality is then

$$0 \le D(g_a \| g_b) \le d(\mathbf{m}^a, \mathbf{m}^b) \tag{7}$$

from which we see that if the distance between $\mathbf{m}^a$ and $\mathbf{m}^b$ is small, the corresponding divergence is small. The distance measure (6) has the useful property that it is symmetric. The distance in (6) has been used with success in speaker clustering applications [10]. From the distance in (6), we can find the corresponding inner product that is the kernel function—converting inner products to distances and vice versa is done via the polar (or polarization) identity [11]. The resulting kernel is

$$\begin{aligned} K(utt_a, utt_b) &= \sum_{i=1}^{N} \lambda_i (\mathbf{m}_i^a)^t \mathbf{\Sigma}_i^{-1} \mathbf{m}_i^b \\ &= \sum_{i=1}^{N} \left(\sqrt{\lambda_i} \mathbf{\Sigma}_i^{-(1/2)} \mathbf{m}_i^a\right)^t \left(\sqrt{\lambda_i} \mathbf{\Sigma}_i^{-(1/2)} \mathbf{m}_i^b\right) \end{aligned} \tag{8}$$

where we have discarded the constant scaling factor for simplicity. The kernel in (8) is linear and involves a simple diagonal scaling of the GMM supervector. Note that since it is linear, it satisfies the Mercer condition [6].

A useful aspect of using the kernel in (8) is that we can use the model compaction technique from [4]. That is, the SVM in (1) can be summarized as

$$f(\mathbf{x}) = \left(\sum_{i=1}^{L} \alpha_i t_i \mathbf{b}(\mathbf{x}_i)\right)^t \mathbf{b}(\mathbf{x}) + d = \mathbf{w}^t \mathbf{b}(\mathbf{x}) + d \tag{9}$$

where $\mathbf{w}$ is the quantity in parenthesis in (9). This means that we only have to compute a single inner product between the target model and the GMM supervector to obtain a score.

V. GMM L^2 Inner Product Kernel

Our second GMM supervector kernel is motivated through the use of function space inner products. Suppose again that we have two GMM models g_a and g_b obtained by MAP adaptation from two utterances utt_a and utt_b. A standard inner product in function spaces is

$$K(utt_a, utt_b) = \int_{R^n} g_a(\mathbf{x}) g_b(\mathbf{x}) d\mathbf{x}. \tag{10}$$

A closed-form solution for the integral in (10) can be found. Using the GMM notation in (3), we obtain

$$\begin{aligned} K(utt_a, utt_b) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \int_{R^n} \mathcal{N}\left(\mathbf{x}; \mathbf{m}_i^a, \mathbf{\Sigma}_i\right) \\ &\quad \times \mathcal{N}\left(\mathbf{x}; \mathbf{m}_j^b, \mathbf{\Sigma}_j\right) d\mathbf{x} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \mathcal{N}\left(\mathbf{m}_i^a - \mathbf{m}_j^b; \mathbf{0}, \mathbf{\Sigma}_i + \mathbf{\Sigma}_j\right) \end{aligned} \tag{11}$$

where $\mathbf{0}$ is the vector of all zeros. Since each of the terms in the sum in (11) is a kernel, and the sum of kernels is also a kernel, then (11) is a kernel (see [6]).

A computationally convenient approximation to (11) is to assume that means from different mixture components are far apart. This makes the terms where $i \neq j$ small in (11). The resulting kernel is

$$\tilde{K}(utt_a, utt_b) = \sum_{i=1}^{N} \lambda_i^2 \mathcal{N}\left(\mathbf{m}_i^a - \mathbf{m}_i^b; \mathbf{0}, 2\boldsymbol{\Sigma}_i\right). \qquad (12)$$

VI. Experiments

We performed experiments on the 2005 NIST speaker recognition (SRE) corpus. For this corpus, we focused on the single-side 8 conversation train, single-side 1 conversation test, English handheld telephone task (the common evaluation condition) [12]. Nominally, the duration of each conversation is 5 min with 2.5 min of speech. This setup resulted in 1672 true trials and 14 406 false trials. We used equal error rate (EER) and the minimum decision cost value (minDCF) as metrics for evaluation. The formula for DCF is

$$\begin{aligned} \mathrm{DCF} = {} & C_{\mathrm{miss}} P(\text{target}) P(\text{miss}|\text{target}) \\ & + C_{\mathrm{FA}} P(\text{nontarget}) P(\text{FA}|\text{nontarget}) \qquad (13) \end{aligned}$$

where $C_{\mathrm{miss}} = 10$, $C_{\mathrm{FA}} = 1$, and $P(\text{target}) = 0.01$.

Our front end processing is as follows. A 19-dimensional MFCC vector is extracted from the pre-emphasized speech signal every 10 ms using a 20-ms Hamming window. The mel-cepstral vector is computed using a simulated triangular filterbank on the DFT spectrum. Bandlimiting is performed by retaining only the filterbank outputs from the frequency range 300–3140 Hz. Cepstral vectors are processed with RASTA filtering. Delta-cepstral coefficients are then computed over a ±2 frame span and appended to the cepstra vector, producing a 38-dimensional feature vector. The feature vector stream is processed through an adaptive, energy-based speech detector to discard vectors from low-energy frames. Feature mapping is then applied to help remove channel effects [13]. Finally, both mean and variance normalization are applied to the individual features.

The GMM UBM consists of 2048 mixture components. For GMM MAP training, we adapt only the means with a relevance factor of 16 [1]. The GMM UBM was trained using EM from the following corpora: Switchboard 2 phase 1, Switchboard 2 phase 4 (cellular), and OGI national cellular.

We produced GMM supervectors on a per-conversation (utterance) basis using MAP adaptation. Both kernels in (8) and (12) were implemented using SVMTorch as an SVM trainer [7]. A background for SVM training consists of GMM supervectors labeled as -1 extracted from utterances from example impostors [4]. An SVM background was obtained by extracting 2326 GMM supervectors from conversations in an English subset of the LDC Fisher corpus.

For enrollment of target speakers, we produced eight GMM supervectors from the eight conversations. We then trained an SVM model using the target GMM supervectors and the SVM background. This resulted in weights and support vector selection from the target speaker and background GMM supervector data sets. For the linear kernel (8), we applied model compaction (9) to obtain a smaller representation. For the L^2 inner product kernel (12), we found it necessary to discard the determinant in the denominator of Gaussian evaluation because of ill-conditioning.

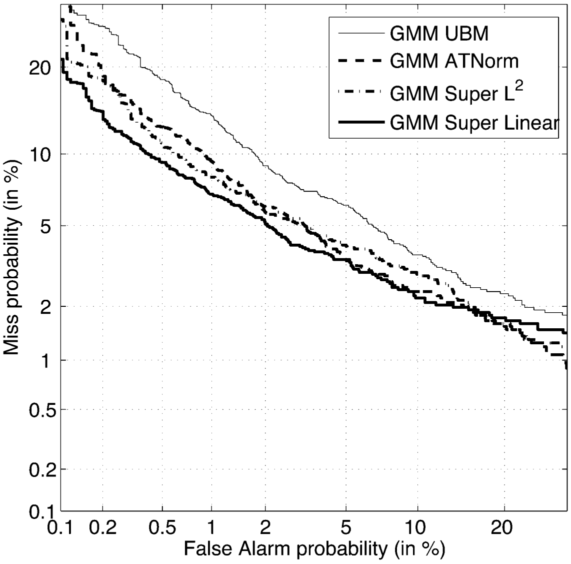

Fig. 2. Comparison of GMM supervector kernels and a standard GMM ATNorm system on the 8 conversation train, 1 conversation test NIST SRE 2005 primary task.

TABLE I
COMPARISON OF EER AND MINDCF FOR DIFFERENT SYSTEMS ON THE 2005 NIST SRE PRIMARY TASK

System	EER	minDCF
GMM UBM	5.68 %	0.0222
GMM ATNorm	4.03 %	0.0172
GMM Super L^2	4.31 %	0.0157
GMM Super Linear	3.77 %	0.0139

Results for the various kernels are shown in Fig. 2 and Table I. In the figure, GMM Super L^2 is the GMM supervector system with the kernel in (12), and GMM Super Linear has the kernel (8). Also, in the figure, we compare the GMM supervector system with standard GMM systems, labeled as GMM UBM and GMM ATnorm. The standard GMM implementation uses the same features as our GMM supervector system. The GMM UBM system is a standard MAP adaptation system with no score normalization. The GMM ATnorm system uses TNorm speakers selected adaptively from the LDC Fisher and Mixer corpora with the method described in [14].

Fig. 2 shows the promise of the new approach. The linear GMM supervector kernel outperforms a standard GMM configuration. This excellent performance is coupled with the fact that the GMM supervector SVM has considerably less computational complexity—no TNorm operation is applied for the GMM supervector system. Additionally, it is well suited for application of new channel compensation techniques such as NAP [15], [16].

VII. Conclusion and Future Work

We have demonstrated two novel kernels for SVMs using GMM supervectors. The SVM was shown to have excellent performance on a NIST SRE 2005 task. Performance was found to be competitive with a standard GMM UBM system with adaptive TNorm. Future work on this method includes applying SVM channel compensation techniques [16] and extending the approach to HMM MAP adaptation.

References

[1] D. A. Reynolds, T. F. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Dig. Signal Process.*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *Proc. Odyssey*, 2004, pp. 219–226.

[3] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Process,*, vol. 13, no. 3, pp. 345–354, May 2005.

[4] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 2002, pp. 161–164.

[5] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," *IEEE Trans. Speech Audio Processing*, vol. 13, no. 2, pp. 203–210, Mar. 2005.

[6] C. Nello and S.-T. John, *Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[7] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *J. Mach. Learn. Res.*, vol. 1, pp. 143–160, 2001.

[8] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback–Leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[9] M. N. Do, "Fast approximation of Kullback–Leibler distance for dependence trees and hidden Markov models," *IEEE Signal Process. Lett.*, vol. 10, no. 4, pp. 115–118, Apr. 2003.

[10] M. Ben, M. Bester, F. Bimbot, and G. Gravier, "Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs," in *Proc. ICSLP*, 2004.

[11] J. B. Conway, *Functional Analysis*. New York: Springer-Verlag, 1990.

[12] (2005) The NIST Year 2005 Speaker Recognition Evaluation Plan. [Online]. Available: http://www.nist.gov/speech/tests/spk/2005/index.htm.

[13] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, 2003, pp. II-53–56.

[14] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for Tnorm in text-independent speaker verification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 2005.

[15] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition," in *Proc Odyssey, Speaker Language Recognition Workshop*, 2004, pp. 57–62.

[16] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 2005, pp. 629–632.