SVM BASED SPEAKER VERIFICATION USING A GMM SUPERVECTOR KERNEL AND NAP VARIABILITY COMPENSATION

W. M. Campbell, D. E. Sturim, D. A. Reynolds, A. Solomonoff

MIT Lincoln Laboratory
Lexington, MA 02420
E-mail: {wcampbell,sturim,dar,als}@ll.mit.edu

ABSTRACT

Gaussian mixture models with universal backgrounds (UBMs) have become the standard method for speaker recognition. Typically, a speaker model is constructed by MAP adaptation of the means of the UBM. A GMM supervector is constructed by stacking the means of the adapted mixture components. A recent discovery is that latent factor analysis of this GMM supervector is an effective method for variability compensation. We consider this GMM supervector in the context of support vector machines. We construct a support vector machine kernel using the GMM supervector. We show similarities based on this kernel between the method of SVM nuisance attribute projection (NAP) and the recent results in latent factor analysis. Experiments on a NIST SRE 2005 corpus demonstrate the effectiveness of the new technique.

1. INTRODUCTION

For the task of text-independent speaker verification, likelihood ratio detection using Maximum A-Posteriori (MAP) adapted Gaussian Mixture Models (GMMs) from a Universal Background model (UBM) has become the standard approach [1]. While this approach provides very good performance, a continuing challenge for robust speaker verification is dealing with channel or session variability (due to microphones, acoustic environments, etc.).

An interesting 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 along with latent factor analysis to perform GMM channel compensation [2].

Support vector machines (SVMs) have proven to be a new effective method for speaker recognition, e.g. [3, 4]. SVMs perform a nonlinear mapping from an input space to an SVM expansion 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 paper, we combine the recent results in SVM methods with the GMM supervector concept. We derive a linear kernel based upon an approximation to KL divergence between two GMM models. We then apply the SVM nuisance attribute projection method [5] to the resulting kernel. We demonstrate similarities between our approach and the latent factor analysis method.

The outline of the paper is as follows. In Section 2, we describe the basic framework for SVMs. In Section 3, we outline the GMM supervector expansion. Section 4 describes the linear kernel for SVM speaker verification. Section 5 discusses the SVM NAP method and relations with latent factor analysis. Finally, in Section 6, we demonstrate the potential of the approach by applying it to a NIST speaker recognition evaluation 2005 task and comparing it to a standard GMM approach.

2. 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, $\sum_{i=1}^{L} \alpha_i t_i = 0$, 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 output 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

[*] This work was sponsored 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 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 SVM expansion space.

The focus of the SVM training process is to model the boundary between classes. For a separable data set, SVM optimization chooses a hyperplane in the expansion space with maximum margin [6]. The data points from the training set lying on the boundaries are the support vectors in equation (1).

3. 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.

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 Figure 1.

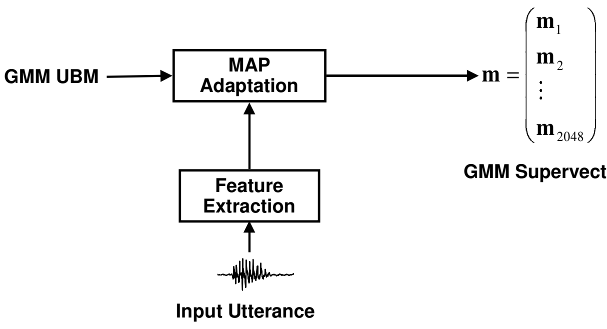
GMM UBM
MAP Adaptation
$\mathbf{m} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_{2048} \end{pmatrix}$
GMM Supervector
Feature Extraction
Input Utterance

Fig. 1. GMM supervector concept

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 [3]—a kernel that directly compares two speech utterances, utt_a and utt_b, and produces $K(utt_a, utt_b)$. 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 $\mathbf{b}(utt_a)$.

4. 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. This results in GMM supervectors, $\mathbf{m}^a$ and $\mathbf{m}^b$. A natural distance between the two utterances is the KL divergence,

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

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],

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

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

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

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 in (6) has been used with success in speaker clustering applications [10]. From the distance in (6), we can find the corresponding inner product which is the kernel function,

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

The kernel in (8) is linear in the GMM supervector; i.e., the mapping from the GMM supervector to SVM expansion space is a diagonal linear transform. Note since it is linear, it satisfies the Mercer condition [6].

A useful aspect of the kernel in (8) is that we can apply the model compaction technique from [3]. 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 we only have to compute a single inner product between the target model and the GMM supervector to obtain a score.

5. SVM NAP AND LATENT FACTOR ANALYSIS

The SVM nuisance attribute projection (NAP) method [5] works by removing subspaces that cause variability in the kernel. NAP constructs a new kernel,

$$\begin{aligned} K(\mathbf{m}^a, \mathbf{m}^b) &= [\mathbf{P}\mathbf{b}(\mathbf{m}^a)]^t \, [\mathbf{P}\mathbf{b}(\mathbf{m}^b)] \\ &= \mathbf{b}(\mathbf{m}^a)^t \mathbf{P}\mathbf{b}(\mathbf{m}^b) \\ &= \mathbf{b}(\mathbf{m}^a)^t (\mathbf{I} - \mathbf{v}\mathbf{v}^t)\mathbf{b}(\mathbf{m}^b) \end{aligned} \tag{10}$$

where $\mathbf{P}$ is a projection ($\mathbf{P}^2 = \mathbf{P}$), $\mathbf{v}$ is the direction being removed from the SVM expansion space, $\mathbf{b}(\cdot)$ is the SVM expansion, and $\|\mathbf{v}\|_2 = 1$. The design criterion for $\mathbf{P}$ and correspondingly $\mathbf{v}$ is

$$\mathbf{v}^* = \operatorname*{argmin}_{\mathbf{v}, \|\mathbf{v}\|_2 = 1} \sum^{i,j} W_{i,j} \|\mathbf{P}\mathbf{b}(\mathbf{m}^i) - \mathbf{P}\mathbf{b}(\mathbf{m}^j)\|_2^2 \tag{11}$$

where the $\{\mathbf{m}^i\}$ are typically a background data set. Here, $W_{i,j}$ can be selected in several different ways. If we have channel nuisance variables (e.g., electret, carbon button, cell) and a labeled background set, then we can pick $W_{i,j} = 0$ when the channels of $\mathbf{m}^i$ and $\mathbf{m}^j$ are the same, and $W_{i,j} = 1$ otherwise. Another criterion is to design the projection based on session variability. In this case, we pick $W_{i,j} = 1$ if $\mathbf{m}^i$ and $\mathbf{m}^j$ correspond to the same speaker, and $W_{i,j} = 0$ otherwise. The idea in both cases is to reduce variability in the SVM kernel distance with respect to nuisances—channel or session.

The solution to (11) is an eigenvalue problem,

$$\mathbf{A}(\operatorname{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W})\mathbf{A}^t\mathbf{v} = \gamma\mathbf{v} \tag{12}$$

where $\mathbf{A}$ is a matrix whose columns are $\mathbf{b}(\mathbf{m}^i)$, $\mathbf{W}$ is the matrix consisting of $W_{i,j}$, and $\mathbf{1}$ is the vector of all ones.

We consider the case of session variability compensation and only one speaker with n sessions; the general case of multiple speakers is a straightforward extension. The SVM NAP problem (12) becomes

$$\mathbf{A}\mathbf{J}\mathbf{A}^t\mathbf{v} = (\mathbf{A}\mathbf{J})(\mathbf{A}\mathbf{J})^t\mathbf{v} = \frac{\gamma}{n}\mathbf{v} \tag{13}$$

where $\mathbf{J} = \mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^t$. The effect of the $\mathbf{J}$ matrix in (13) is to replace every vector $\mathbf{m}^i$ by how much it deviates from the average vector across all sessions, $\bar{\mathbf{b}}$. The operation $(\mathbf{A}\mathbf{J})(\mathbf{A}\mathbf{J})^t$ is just an autocorrelation. Thus, for this particular case of NAP, we are finding the principal component of the autocorrelation matrix of the vectors $\mathbf{A}\mathbf{J}$,

$$(\mathbf{A}\mathbf{J})(\mathbf{A}\mathbf{J})^t = \sum_{i=1}^{n} (\mathbf{b}(\mathbf{m}^i) - \bar{\mathbf{b}})(\mathbf{b}(\mathbf{m}^i) - \bar{\mathbf{b}})^t \tag{14}$$

In the latent factor analysis method of Kenny [2], the GMM supervector $\mathbf{m}(s, i)$ is dependent on the speaker s and the session i. The supervector is a sum of a speaker component and a session dependent vector,

$$\mathbf{m}(s, i) = \mathbf{m}(s) + \mathbf{U}\mathbf{n}(s, i). \tag{15}$$

The latent factor $\mathbf{n}(s, i)$ is assumed to be zero mean, unit variance and Gaussian. For a large number of sessions, this means

$$\bar{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{m}(s, i) \approx \mathbf{m}(s). \tag{16}$$

So, we can derive the subspace generated by $\mathbf{U}$ by finding the principal components of the autocorrelation matrix

$$\mathbf{R} = \sum_{i=1}^{n} (\bar{\mathbf{m}} - \mathbf{m}(s, i))(\bar{\mathbf{m}} - \mathbf{m}(s, i))^t. \tag{17}$$

Comparing equations (14) and (17), we see that NAP with the linear kernel, $\mathbf{b}(\mathbf{m}^i) = \mathbf{m}^i$, and session variability as a nuisance variable (13) produces the same subspace as latent factor analysis.

The relation between NAP and factor analysis opens up new possiblities. First, note that the SVM NAP method of using the kernel matrix [5] can be applied to solve for the subspace generated by $\mathbf{U}$. Second, for a nonlinear kernel [11], SVM NAP uses a nonlinear expanded version of the GMM supervector producing a variability compensation method distinct from the linear method presented here. Third, note that the method of using the subspace is different between SVM NAP and latent factor analysis. For SVM NAP, the subspace is removed from the GMM supervector by projection. For Vogt's latent factor analysis [12], an iterative method is used to estimate the latent variables, and then the variability is subtracted from the GMM supervector. Further work is needed to understand the advantages of these different approaches.

6. EXPERIMENTS

We performed experiments on the 2005 NIST speaker recognition (SRE) corpus. We focused on the single-side 8 conversation train, single-side 1 conversation test, English handheld telephone task (the common evaluation condition) [13]. This setup resulted in $1,672$ true trials and $14,406$ false trials.

For feature extraction, a 19-dimensional MFCC vector is found from pre-emphasized speech every 10 ms using a 20 ms Hamming window. Delta-cepstral coefficients are computed over a ± 2 frame span and appended to the cepstra producing a 38 dimensional feature vector. An energy-based speech detector is applied to discard vectors from low-energy frames. To mitigate channel effects, RASTA, feature mapping, and mean and variance normalization are applied to the 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. The kernel in equation (8) was 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 [3]. An SVM background was obtained by extracting $2,326$ GMM supervectors from conversations in an English subset of the LDC Fisher corpus.

For enrollment of target speakers, we produced 8 GMM supervectors from the 8 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 SVM NAP, a corank 64 projection was designed based on session variability, see Section 5. The projection was trained with Switchboard 2 parts 1, 4, and 5. This projection was applied to the SVM background and to all training utterances. No projection was needed in scoring since $\mathbf{P}^2 = \mathbf{P}$.

Results for the various approaches are shown in Figure 2. In the figure, GMM Super Linear has the kernel (8). GMM Super NAP 64 demonstrates applying the SVM NAP method to the GMM supervector linear kernel. Also, in the figure, we compare the GMM supervector system with two 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]. For the GMM Super NAP 64 system, the equal error rate (EER) is 3.13%, and the NIST minimum decision cost function value is 0.010.

Figure 2 shows the promise of the new approach. The linear GMM supervector kernel outperforms a standard GMM configuration for low false alarm rates and at EER. 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.

7. CONCLUSIONS

We have demonstrated a novel kernel for SVMs using GMM supervectors. The SVM was shown to have excellent performance on a NIST SRE 2005 task and to outperform standard GMM systems. Application of session variability compensation with SVM NAP improved performance substantially. Future work includes exploring other GMM supervector kernels and applying these methods to HMM MAP adaptation.

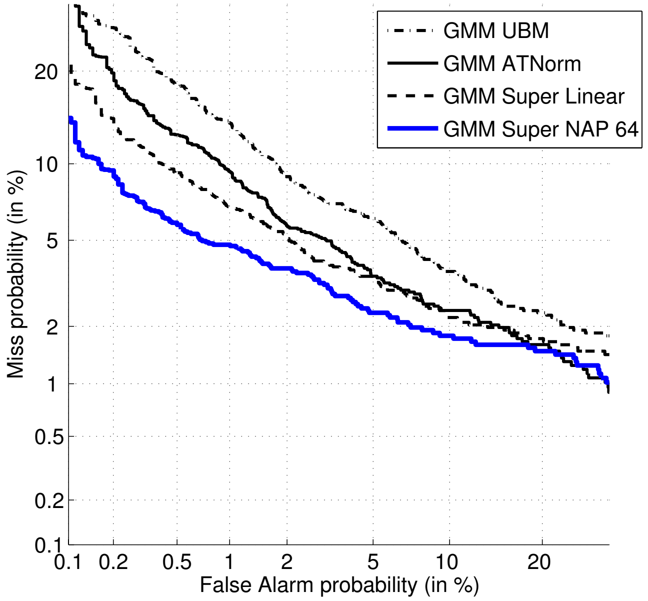

Fig. 2. A comparison of GMM supervector methods with standard GMM UBM and ATNorm systems on an 8 conversation train, 1 conversation test NIST SRE 2005 task.

8. REFERENCES

[1] 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.

[2] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *Proc. Odyssey04*, 2004, pp. 219–226.

[3] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of ICASSP*, 2002, pp. 161–164.

[4] V. Wan and S. Renals, "SVMSVM: support vector machine speaker verification methodology," in *Proceedings of ICASSP*, 2003, pp. 221–224.

[5] Alex Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proceedings of ICASSP*, 2005.

[6] Nello Cristianini and John Shawe-Taylor, *Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[7] Ronan Collobert and Samy Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[8] Pedro J. Moreno, Purdy P. Ho, and Nuno Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," in *Adv. in Neural Inf. Proc. Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.

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

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

[11] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *submitted to IEEE Signal Processing Letters*, 2005.

[12] Robbie Vogt, Brendan Baker, and Sridha Sriharan, "Modelling session variability in text-independent speaker verification," in *Proc. Interspeech*, 2005, pp. 3117–3120.

[13] "The NIST year 2005 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/2005/index.htm, 2005.

[14] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for Tnorm in text-independent speaker verification," in *Proceedings of ICASSP*, 2005.