Support Vector Machines Based Text Dependent Speaker Verification Using HMM Supervectors

Chengyu Dong [1], *Yuan Dong* [1, 2], *Jing Li* [2], *Haila Wang* [1]

[1] France Telecom R&D Beijing Co, Ltd., Beijing, P. R. China
[2] School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing, P. R. China

{chengyu.dong,haila.wang}@orange-ftgroup.com, yuandong@bupt.edu.cn

Abstract

Conventional subword based hidden Markov models (HMMs) have proven to be an effective approach for text-dependent speaker verification. The standard training method works by modeling the MAP adapted means of subword HMMs. In this paper, we propose the use of HMM supervectors from the speaker models as features in support vector machines (SVMs) classifier. An HMM supervector is constructed by stacking means of adapted mixture components from all states within HMMs. We present two SVM kernels: linear kernel and dynamic time alignment kernel (DTAK) based on the KL divergence to evaluate the system. In addition, another effective method is proposed to normalize SVM output scores using speaker independent HMM supervectors. Experimental results show that the SVM system with HMM supervectors achieves lower performance than conventional HMM verification system, but their fusion can give a significant improvement.

Index Terms: SVM, HMM supervectors, DTAK, fusion

1. Introduction

Speaker recognition is an important, emerging technology with many potential applications. Due to the constraint that enrollment and testing utterance have the same content, text-dependent speaker verification (TDSV) systems can achieve better performance than text-independent systems. Recently TDSV systems with user-customized password [1] have become a promising speaker authentication solution, where users can choose his/her own password from an unconstrained vocabulary. The standard approach to this problem is to use subword based hidden Markov models (HMMs) [2, 3, 4].

Support vector machines (SVMs) have become popular these years. SVMs work on a high-dimensional expansion space which is gained by a nonlinear mapping from the input space. Many studies have shown that SVM based speaker verification can remarkably improve the performance [5, 6, 7]. SVMs with GMM supervectors have proven to be an effective method for text-independent tasks. This key innovation in this approach is to use a GMM supervector consisting of the stacked means of the mixture components [8, 9]. When migrating SVM to HMM based systems, there are some issues to solve, e.g. sparse data and mismatch between two dynamic patterns. DTAK [10] is evaluated to handle the latter problem for speech recognition task.

In this paper, we firstly borrow the concept from GMM supervectors [8] and expand the idea to subword based HMM models. We show the method for finding an approximation to KL divergence between two HMM supervectors. From our study, the distance correlates with the state transition matrix and multivariate Gaussian mixtures within the HMM states. To accord with the KL divergence, we use both linear kernel and DTAK to evaluate. Additionally, another normalization method is proposed for the lack of training data. Fusion with this new system with conventional HMM and GMM is also investigate in this study.

The remainder of this paper is organized as follows. In Section 2, we will review the conventional subword HMM based speaker verification. Section 3 will present the use of HMM supervectors in SVM based speaker verification. In Section 4, we propose the method to normalized SVM output scores. Finally the experimental results are given in Section 5 at the end of this paper.

2. HMM Baseline System

This section describes the traditional text-dependent speaker verification system. It involves two kinds of sessions, enrollment and verification. In enrollment session, a speaker ***S*** is asked to repeat a pre-selected spoken pass-phrase ***P*** for several times. These enrollment data are then used to adapt the background speaker-independent (SI) HMM model λ^{b} using maximum a posteriori (MAP) adaptation. A speaker-dependent (SD) HMM model λ^{s} is then constructed to represent both speaker ***S*** and pass-phrase ***P***.

During verification session, we assume that ***P*** is compose of a sequence string of N subwords, $S_1, S_2, ..., S_N$, where N is the total number of ***P***. Given a test utterance $O = \{O_1, O_2, ..., O_T\}$, speaker verification decides whether O is produced by the claimed speaker. We need a decision rule to give reliable verification scores. Under the Neyman-Pearson lemma, the log likelihood ratio (LLR) is given by:

$$\Lambda(O;\mathrm{S}) = \log P\left(O_1^T \mid \lambda^s\right) - \log P\left(O_1^T \mid \lambda^b\right) \qquad (1)$$

In subword based text-dependent speaker verification, the input utterance is firstly segmented into N phones using speaker independent models. This process is called 'forced alignment'. Then the observation sequence can be regard as N segments $O = \left\{O_1^{t_1}, O_{t_1+1}^{t_2}, ..., O_{t_{N-1}+1}^{t_N}\right\}$, where frame $t_{i-1}+1$ to frame t_i are belonging to the *i*th phone. The LLR of *i*th segment can be denoted as:

$$\Lambda_i\left(O_{t_{i-1}+1}^{t_i}; S_i\right) = \log P\left(O_{t_{i-1}+1}^{t_i} \mid \lambda^s; S_i\right) - \log P\left(O_{t_{i-1}+1}^{t_i} \mid \lambda^b; S_i\right) \qquad (2)$$

As the result, the final verification score can be simply computed as follows:

$$\Lambda(O;\mathrm{S}) = \frac{1}{\mathrm{T}} \sum_{i=1}^{N} \Lambda\left(O_{t_{i-1}+1}^{t_i}; S_i\right) \qquad (3)$$

3. SVM Based Speaker Verification Using HMM Supervectors

3.1. HMM supervectors

HMMs are generative models which can offer more flexibility for text-dependent speaker verification. One HMM model consists of several states. Each state can be denoted as multivariate Gaussian mixture functions, we have:

$$b_j(x) = \sum_{k=1}^{M} c_{jk} N(x; m_{jk}, \Sigma_{jk}) \tag{4}$$

where $N(x; m_{jk}, \Sigma_{jk})$ denotes a single Gaussian density function with mean vector m_{jk} and covariance matrix Σ_{jk} for state j, M is the number of mixtures and c_{jk} is the mixture weight for ***k***th component.

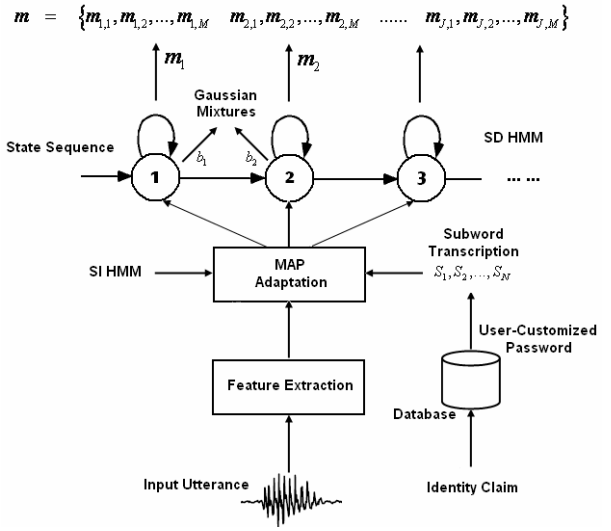

Figure 1: *The block diagram of HMM supervectors extraction*

Fig. 1 illustrates the process of HMM supervector extraction. After the speaker claims the identity, the system can obtain the transcription and construct a concatenated subword string. Given an input utterance, SD HMMs training is performed by MAP adaptation of only the means m_{jk} to SI HMMs. SI HMM model plays the same role as the universal background model (UBM) [11] in text independent speaker verification. Then we form the supervectors $m_j = \{m_{j1}, m_{j2}, ..., m_{jM}\}$ of state j from the adapted HMM model. Consequently the supervectors from the whole state sequence will form the concatenated HMM supervectors $m = \{m_1, m_2, ..., m_J\}$, where J is the total number of states within a subword string.

The HMM supervector can be expressed as a mapping between an HMM model and a high-dimensional vector. For the case of linear kernel, the mapping is from an HMM model λ^s to HMM supervectors m^s.

3.2. Support Vector Machines

Support Vector Machines are state-of-the-art tools for classification tasks. In speaker verification, SVM can be treated as a two-class classifier given by:

$$f(v) = \sum_{i=1}^{L} \alpha_i y_i K(v_i, v) + d \tag{5}$$

where $K(v_i, v_j)$ is a kernel function, d represents a possible bias. The $y_i \in \{\pm 1\}$ is the actual targets, with respect to α_i and subject to $\sum_{j=1}^{L} \alpha_i y_i = 0$. All of the parameters can be obtained through the training set that maximizes the margin between two classes.

As a kernel function $K(v_i, v_j)$, linear kernel is used in GMM supervectors as follows:

$$K(v_1, v_2) = \phi(v_1)^T \phi(v_2) \tag{6}$$

where $\phi(\bullet)$ is a mapping function from input feature space to SVM dimensional expansion space.

3.3. HMM Supervectors Linear Kernel

Suppose we have two SD models λ^{a} and λ^{b} which are produced by speaker a and speaker b respectively. With a given pass-phrase, the HMM supervectors can be extracted by the subword string and the SD HMMs. A commonly used measurement of the distance between two HMM models is the Kullback-Leibler divergence (KLD) which is defined as:

$$D(\lambda^{\mathrm{a}} \| \lambda^{\mathrm{b}}) = \int_{R^x} \lambda^{\mathrm{a}}(x) \log \frac{\lambda^{\mathrm{a}}(x)}{\lambda^{\mathrm{b}}(x)} dx \tag{7}$$

To approximate the KLD for HMMs, the idea is to estimate the upper bound using the log-sum inequality. The result is given by [12]:

$$D(\lambda^{\mathrm{a}} \| \lambda^{\mathrm{b}}) \le \sum_{j=1}^{J} \xi_j \left(D(a_j^{\mathrm{a}} \| a_j^{\mathrm{b}}) + D(b_j^{\mathrm{a}} \| b_j^{\mathrm{b}}) \right) \tag{8}$$

The upper bound in (8) can be computed directly using the model parameters $\lambda = (\mathrm{A}, \mathrm{B}, \Pi)$, where $\mathrm{A} = \{\alpha_{\mathrm{ij}}\}$ to be the state-transition probability matrix: $\alpha_{i,j} = P(S_t = j \mid S_{t-1} = i)$. Here we denote the discrete probability function of state j: $a_j = \{\alpha_{j,1}, ..., \alpha_{j,J}\}$. $b_j(\bullet)$ can be a parameterized pdf which is defined in (4). There exists a distribution vector ξ_j such that $\xi_j^T \mathrm{A} = \xi_j^T$.

In text-dependent speaker verification, a constrained CDHMM model which has a time sequence structure is used. The constraints imposed on the model allow for transitions

from one state to the following state and to itself. By assuming that $\alpha_{j,i}=0\left(i\neq j, i\neq j+1\right)$, $\alpha_{j,j+1}=1-\alpha_{j,j}$, we have $\xi_j=\mathrm{I}$, and the KLD between two state-transition probability function becomes:

$$D\left(a_j^{\mathrm{a}}\,\|\,a_j^{\mathrm{b}}\right)=\alpha_{jj}^{\mathrm{a}}\log\left(\frac{\alpha_{jj}^{\mathrm{a}}}{\alpha_{jj}^{\mathrm{b}}}\right)+\left(1-\alpha_{jj}^{\mathrm{a}}\right)\log\left(\frac{1-\alpha_{jj}^{\mathrm{a}}}{1-\alpha_{jj}^{\mathrm{b}}}\right) \quad (9)$$

Since b_j^{a} and b_j^{b} are mixtures of Gaussians, we can estimate their KLD between two M-component Gaussians:

$$\begin{aligned} D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right) &= D\left(N\left(\cdot;m_j^{\mathrm{a}},\Sigma_j^{-1}\right)\|\,N\left(\cdot;m_j^{\mathrm{b}},\Sigma_j^{-1}\right)\right) \\ &= \sum_{k=1}^{M} c_{jk}\left(m_{jk}^{\mathrm{a}}-m_{jk}^{\mathrm{b}}\right)\Sigma_{jk}^{-1}\left(m_{jk}^{\mathrm{a}}-m_{jk}^{\mathrm{b}}\right) \end{aligned} \quad (10)$$

In HMM based speaker verification system, the influence of KLD between two state-transition probability function is much less than the influence of the KLD between two multivariate Gaussians. Therefore we obtain:

$$\frac{D\left(a_j^{\mathrm{a}}\,\|\,a_j^{\mathrm{b}}\right)}{D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right)}\ll 1 \quad (11)$$

By applying (9) (10) (11), (8) becomes:

$$\begin{aligned} D\left(\lambda^{\mathrm{a}}\,\|\,\lambda^{\mathrm{b}}\right) &\leq \sum_{j=1}^{J}\left(D\left(a_j^{\mathrm{a}}\,\|\,a_j^{\mathrm{b}}\right)+D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right)\right) \\ &\leq \sum_{j=1}^{J} D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right)\left(1+\frac{D\left(a_j^{\mathrm{a}}\,\|\,a_j^{\mathrm{b}}\right)}{D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right)}\right) \\ &\approx \sum_{j=1}^{J} D\left(b_j^{\mathrm{a}}\,\|\,b_j^{\mathrm{b}}\right) \end{aligned} \quad (12)$$

By taking (10), the approximate KL distance of these two HMMs can be formulated as:

$$d\left(\mathrm{a},\mathrm{b}\right)=\frac{1}{2}\sum_{j=1}^{J}\sum_{k=1}^{M} c_{jk}\left(m_{jk}^{\mathrm{a}}-m_{jk}^{\mathrm{b}}\right)\Sigma_{jk}^{-1}\left(m_{jk}^{\mathrm{a}}-m_{jk}^{\mathrm{b}}\right) \quad (13)$$

Using the concept of GMM supervector, we can convert inner products to distance. Therefore, from the distance given above, the kernel function can be denoted as the corresponding inner product between two supervectors constructed from their respective SD HMMs. Here we have:

$$\begin{aligned} K\left(\nu_{\mathrm{a}},\nu_{\mathrm{b}}\right) &= \sum_{g=1}^{J\times M}\left(\sqrt{c_g}\,\Sigma_g^{-\frac{1}{2}}m_g^{\mathrm{a}}\right)^{T}\left(\sqrt{c_g}\,\Sigma_g^{-\frac{1}{2}}m_g^{\mathrm{b}}\right) \\ &= \phi\left(\nu_{\mathrm{a}}\right)^{T}\phi\left(\nu_{\mathrm{b}}\right) \end{aligned} \quad (14)$$

where we discard the constant scaling factors. The subscript g represents $j\cdot M+k$ which is mapping from the mean matrix $\left\{m_{jk}\right\}$ to the mean vector $\left\{m_g\right\}$. The kernel in (14) is linear and the expansion from HMM supervectors to SVM higher dimension space.

Considering (11) and the principle of MAP adaptation, we use supervectors only from b_j to construct linear kernel. Another reason why we didn't use full approximation in (8) instead of the simple one in (12) is that the sum of two KLDs in (8) can not be expressed as the symmetric form of inner product between two supervectors. For simplicity, we could only use concatenated supervectors from multivariate Gaussians of each state. However we might lose some information on state duration using SVM with linear kernel.

3.4. HMM Supervectors Dynamic Time Alignment Kernel

The remainder term $D\left(a_j^{\mathrm{a}}\,\|\,a_j^{\mathrm{b}}\right)$ of (8) specifies the state transition which expresses the time dependency of HMM models. In order to exploit this information more efficiently, our attention was diverted by time normalizing kernels in SVMs. Therefore the standard solution based on dynamic time alignment kernel [10] is proposed to solve this issue. The basic idea of nonlinear time alignment is incorporated into the kernel function.

Assume we have two sequence vectors X and Y. These two patterns have equal lengths. The inner product between X and Y can be calculated directly. We have:

$$K_{LIN}\left(X,Y\right)=X\circ Y=\frac{1}{L}\sum_{j=1}^{L} x_j^{T}\cdot y_j \quad (15)$$

Unlike the linear kernel, DTAK can be performed in such way that maximizes the accumulated similarity:

$$K_{DTA}\left(X,Y\right)=X\odot Y=\max_{\psi_I,\psi_J}\frac{1}{M_\psi}\sum_{j=1}^{L}\omega\left(j\right)x_{\psi_I(j)}^{T}\cdot y_{\psi_J(j)} \quad (16)$$

Subject to

$$1\leq\psi_I\left(j\right)\leq\psi_I\left(j+1\right)\leq\left|X\right| \quad (17)$$

$$1\leq\psi_J\left(j\right)\leq\psi_J\left(j+1\right)\leq\left|Y\right| \quad (18)$$

where $\omega\left(j\right)$ is a path weighting coefficient, and M_ψ is a normalizing factor. In our study, $M_\psi=\left|X\right|+\left|Y\right|$. Then the optimization problem is solved using the following recursive equation by means of dynamic programming:

$$D\left(i,j\right)=\max\begin{cases} D\left(i,j-1\right)+x_i^{T}\cdot y_j \\ D\left(i-1,j-1\right)+2\cdot x_i^{T}\cdot y_j \\ D\left(i-1,j\right)+x_i^{T}\cdot y_j \end{cases} \quad (19)$$

This form of recursive function is symmetric, so that the kernel can follow max-margin criterion. It is worth mentioning that in contrast with linear kernel, DTAK function requires about $\left|X\right|^2$ operations, while the linear kernel only needs $\left|X\right|$ operations. This difference leads directly to much higher

computational cost with DTAK kernel than the cost with linear kernel.

Applying the DTA kernel to SVM system using HMM supervectors, we have:

$$K_{DTA}(\nu_{\mathrm{a}},\nu_{\mathrm{b}})=\max_{\psi_I,\psi_J}\frac{1}{M_\psi}\sum_{j,k}\omega(j)\left(\phi^{\mathrm{a}}_{\psi_I(j)k}\right)^T\cdot\left(\phi^{\mathrm{b}}_{\psi_J(j)k}\right) \quad (20)$$

$$\begin{aligned} where:\ \phi^{\mathrm{a}}_{\psi_I(j)k}&=\sqrt{c_{\psi_I(j)k}}\,\Sigma^{-\frac{1}{2}}_{\psi_I(j)k}m^{\mathrm{a}}_{\psi_I(j)k},\\ \phi^{\mathrm{b}}_{\psi_J(j)k}&=\sqrt{c_{\psi_J(j)k}}\,\Sigma^{-\frac{1}{2}}_{\psi_J(j)k}m^{\mathrm{b}}_{\psi_J(j)k}\end{aligned} \quad (21)$$

HMM supervectors with DTAK can be seen as a kind of kernel normalization technique which handles the problem of state alignment in HMM. In our MAP adaptation process, the phone segment can be fixed using Baum-Welsh algorithm until the convergence is obtained. In contrast to triphone model approach used in the hybrid HMM/SVM system [13], monophone based HMMs might not align the HMM states corresponding to the utterance segment so well. Therefore we use DTA kernel instead of linear kernel.

4. Normalized SVM Scores using Speaker Independent HMM Supervector

Using the HMM supervector introduced in the last section, the SVM discriminant function in (5) can be summarized as:

$$S=f\left(\nu^s\mid \mathrm{M}^s\right)=\left(\sum_{j=1}^{J}\alpha_j y_j\phi(\nu_j)\right)^T\phi\left(\nu^s\right)+d=W^T\phi\left(\nu^s\right)+d \quad (22)$$

where W denotes the optimum decision boundary from the training data, M^s is the SVM model of speaker ***s***.

The concept of normalizing SVM score comes from zero normalization (Z-Norm) [14]. A speaker SVM model is tested against a background SI HMM supervector and the output SVM score is used to normalize the score from testing utterance. The normalization has the form:

$$\bar{S}=f\left(\nu^s\mid \mathrm{M}^s\right)-f\left(\nu^b\mid \mathrm{M}^s\right) \quad (23)$$

where the HMM supervector ν^b derives from the background SI HMMs. The advantage of this form is that the normalization parameter $f\left(\nu^b\mid \mathrm{M}^s\right)$ can be performed off-line during training. Unlike Z-Norm techniques, the variance will not be applied.

The main reason why we used SI HMM supervector to normalize the SVM output score is the lack of training data to adapt. Generally speaking, we produced an HMM supervector on a per-utterance basis using MAP adaptation. Some Gaussian components might not be adapted. These dimensions remain the property of SI HMMs. Therefore, this part of the supervector has no discrimination. To explain the reason more precisely, we partition an HMM supervector into two parts:

$$\phi\left(\nu^s\right)=\left(\phi\left(\tilde{\nu}^s\right),\phi\left(\nu^b\right)\right) \quad (24)$$

where $\phi\left(\tilde{\nu}^s\right)$ denotes the dimensions which are adapted, $\phi\left(\nu^b\right)$ is the remaining part of SI HMM means.

The corresponding W is also divided into W_1^T for weighting $\phi\left(\tilde{\nu}^s\right)$, and W_2^T for weighting $\phi\left(\nu^b\right)$:

$$W^T=\left(W_1^T,W_2^T\right) \quad (25)$$

Here we have:

$$\begin{aligned} S&=W^T\phi\left(\nu^s\right)+d\\ &=\underline{W_1^T\phi\left(\tilde{\nu}^s\right)}+\underline{\underline{W_2^T\phi\left(\nu^b\right)}}+d\end{aligned} \quad (26)$$

When we use normalized SVM scores $\bar{S}$ instead of S, the form (23) becomes:

$$\begin{aligned} \bar{S}&=\left(W^T\phi\left(\nu^s\right)+d\right)-\left(W^T\phi\left(\nu^b\right)+d\right)\\ &=W_1^T\left(\phi\left(\tilde{\nu}^s\right)-\phi\left(\nu^b\right)\right)+W_2^T\left(\phi\left(\nu^b\right)-\phi\left(\nu^b\right)\right)\\ &=\underline{W_1^T\left(\phi\left(\tilde{\nu}^s\right)-\phi\left(\nu^b\right)\right)}\end{aligned} \quad (27)$$

Comparing with these two forms, we can find that the part with a single underline has discriminative ability, while the one with double underline has no discrimination. From the SVM theory, the form $\phi\left(\tilde{\nu}^s\right)-\phi\left(\nu^b\right)$ can only shift the input feature space, but can't change the separating hyper-plane in such a high-dimensionality space. Therefore $\bar{S}$ is more discriminative than S. Beside this main reason, SVM score normalization can provide more stable speaker specific threshold like Z-Norm.

5. Experimental Results

5.1. Corpora and Front-end

We now present the results of SVM based text-dependent speaker verification using HMM supervectors. We evaluated the proposed algorithm here in the telephone based Mandarin speaker verification database which consists of 214 speakers (101 males and 113 females). 80 speakers form development set which was used to adjust the parameters. The rest 134 speakers were used for evaluation. The evaluation data was divided into two groups. One group had 54 target speakers and the other group had 80 imposter speakers. The target trials were evaluated in the first group, while the imposter trials were performed using imposter group to impose against the target speakers. The lengths of selected 10 fixed passwords vary from 5 to 10 Chinese characters. The imposters were assumed to know the exact password of the target speaker. In the experiment, each target speaker was required to say the same passwords for 12 times over an interval of one week. Two utterances of the pass-phrase recorded from two separate sessions were used for enrollment. The remaining 10 sessions were used for verification. The latest session might be recorded three months after the enrollment session. So total 54×10×10=5400 utterances made the set of target trials, and 80×10×10=8000 selected phrases compose the imposter trials.

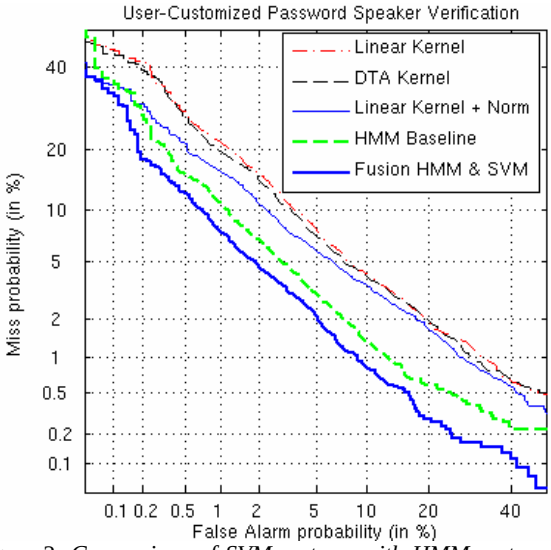

Figure 2: *Comparison of SVM systems with HMM systems in user customized password speaker verification*

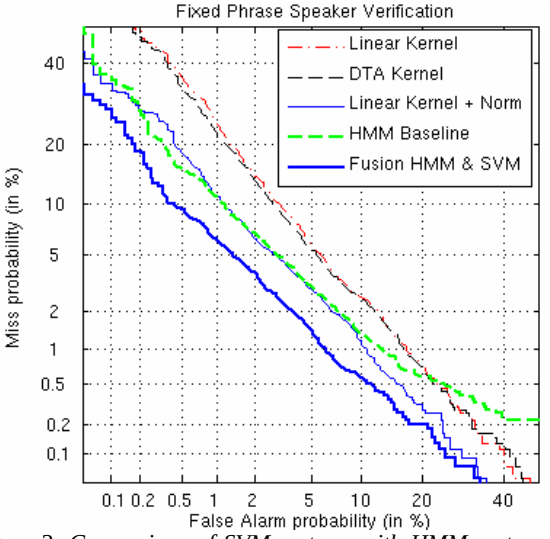

Figure 3: *Comparison of SVM systems with HMM systems in fixed phrase speaker verification*

For all experiments in this paper, we first ran the recognizer using utterance verification [15] on the entire corpus. Then we removed 2% utterances that were contaminated from the telephone channel using utterance verifier. Finally in the resulting cut corpus, we had total 5292 target trials and 7840 imposter trials.

The acoustic features used in our system are the first 12 perceptual linear prediction (PLP) coefficients together with the log-energy of each frame which are calculated every 10 ms using a 25ms Hamming window. The features are processed through a RASTA channel equalization filter. By including the first and the second derivatives over ±2 frame span, 39-dimensional feature vectors were finally used.

5.2. Evaluations

To evaluate the TDSV system, we use decision error tradeoff (DET) curves which have been widely used for representation of detection task performance.

Another important evaluation factor is minimum detection cost function (DCF) which is defined:

$$DCF = C_{FR} \cdot P_{FR} \cdot P_{Target} + C_{FA} \cdot P_{FA} \cdot \left(1 - P_{Target}\right) \qquad (28)$$

where P_{Target} is a priori probability of target tests with $P_{Target} = 0.01$. And the specific cost factors $C_{FR} = 10$ and $C_{FA} = 1$. So the interest is shifted to low FA rates.

5.3. SVM Training

An important issue that comes up with new HMM supervectors is how to train an SVM model using our proposed method. For any two utterances from two speakers, the same subword sequence may not be spoken. In our system framework, we could only assign the SI HMM supervector mean to evaluate the kernel. Another difficulty that arises when applying SVM in user-customized password speaker verification is the lack of imposter utterances. Therefore we must seek for other utterance segments which are belonging to the specific subwords to train the imposter SD HMM model. Then the supervectors were formed using SD HMMs. Consequently we pool all the supervectors of imposters and the target speaker together to compose the training material.

We use a set of context-independent (CI) phone units as a universal phone set. There are 60 (21 initials, 38 finals and 1 silence) CI phoneme models. The model was trained on about 90 hours of data with 282 speakers. The optimal number of mixtures per state is determined to 16 empirically. For HMM MAP adaptation, the relevance factor was set to 1. The kernels in (14) and (20) were implemented with SVMTorch [16]. The background speakers came from the corpus of SI HMM training data. Each speaker in this corpus was required to speak 50 phrases. So the imposter SD HMMs can be constructed by adapting SI HMMs. For enrollment of target speakers, we produced two HMM supervectors from two separate sessions. We then trained an SVM model using the target HMM supervectors and the background supervectors. After the SVM training process, the weights and the support vectors can be obtained from the target speaker and the background speakers.

Considering the fixed-phrase speaker verification task, we can obtain the imposter utterances which have the same transcriptions as the target speakers. In this condition, SD HMMs are directly built from these imposter utterances.

5.4. System Fusion

The motivation for the system fusion is that a combined verification system can significantly outperform over the individual approaches. The baseline HMM score, $\Lambda(O;\mathrm{S})$, and the SVM score, $f(O;\mathrm{S})$, are computed for all test utterances. Then the combined verification score is given by:

$$F = \eta \cdot \Lambda(O;\mathrm{S}) + (1-\eta) \cdot f(O;\mathrm{S}) \qquad (29)$$

where η is a weighting factor determined as part of training phase. The weight is determined through a discriminant analysis procedure [17] like LDA which follows the Fisher's discrimination criterion. The weight was tuned from the development set.

5.5. Experimental Results

Text dependent speaker verification experiments were conducted, and the results are summarized in Fig.2 and Fig. 3 for user-customized password speaker verification and fixed-phrase speaker verification respectively. These results correspond to the same evaluation data but different imposter HMM supervectors when training the SVM model. Both figures show the fact that DTA kernel performs better than linear kernel, but no significant improvement. Additionally the computational cost of DTAK is much higher than linear kernel. Therefore we didn't adopt DTAK to perform with normalization. After applying normalization to SVM output scores, the performance of the proposed SVM based system can be improved, especially in fixed phrase speaker verification.

Results for the combination of HMM and SVM are shown in both figures, the fusion system of both HMM and SVM can remarkably improve the performance. The equal error rate (EER) drops from 4.01% for the HMM to 3.47% for the HMM/SVM fusion system in user-customized password speaker verification system. In addition, the EER is reduced to 2.71% in fixed phrase speaker verification system, a 32% improvement. From these two figures, we can conclude that:

1. The DTA kernel performs a little better than the linear kernel, but requires too much computational cost. This explains why we discard the KLD form $D\left(a_j^{\mathrm{a}} \| a_j^{\mathrm{b}}\right)$. DTAK preserves the state duration information. But HMM can't provide a good model of duration. DTAK couldn't improve performance greatly.
2. Normalized output score can remarkably improve the performance of the SVM system. In fixed phrase speaker verification system, the normalization function is more efficient. The reason can be explained that in such system, the training data is more insufficient. So the normalization function can outperform well when comparing to user-customized system.
3. However, HMM baseline performs better than SVM system. When fusing both systems together to a combined one, we can achieve the best performance.

5.6. Discussion with HMMs, GMMs and SVMs

To further improve our system, we combined HMM, GMM and SVM systems together. We show comparison of the proposed SVM system with the traditional HMM and GMM based ones. In GMM based systems, the two enrollment utterances were treated as text-independent. 128 mixture components compose the UBM which were trained using EM. For GMM MAP training, we adapt only the means. So the testing utterance will be evaluated as text-independent. Fig. 4 gives the performance of individual systems and their combinations. We can see from this figure, any two combinations can remarkably improve the performance. When we fuse all the three systems together, we can achieve the best result (EER=2.95%), 9.0% relative improvement than HMM/GMM combination (EER=3.24%) in terms of EER.

Fig. 5 illustrates the 2-dimensional distributions of the scores derived from HMM and SVM classifiers for target speakers and imposter speakers. We can see from this figure, the correlation of the two classifiers is very low making the target trails scores dispersed. That explains why the fusion system can improve the performance of individual systems. The 3-D score distribution is shown from Fig. 6 adding GMM systems to the third dimension. From the two figures, we can obtain more discriminative information from the three individual systems. Therefore we can achieve best fusion performance.

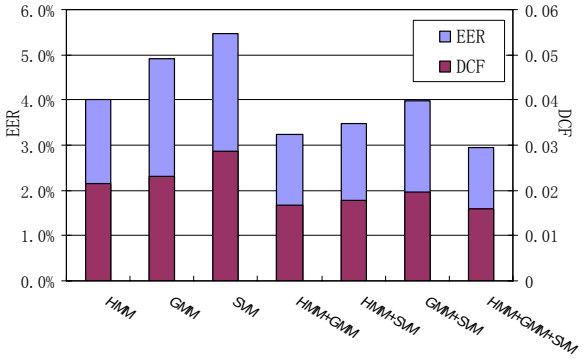

Figure 4: *System fusions on HMMs, GMMs and SVMs*

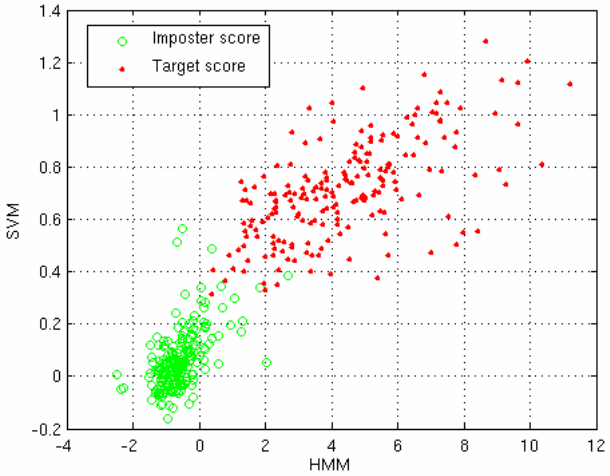

Figure 5: *2-D distribution of the scores for target and imposter trials (HMM and SVM scores)*

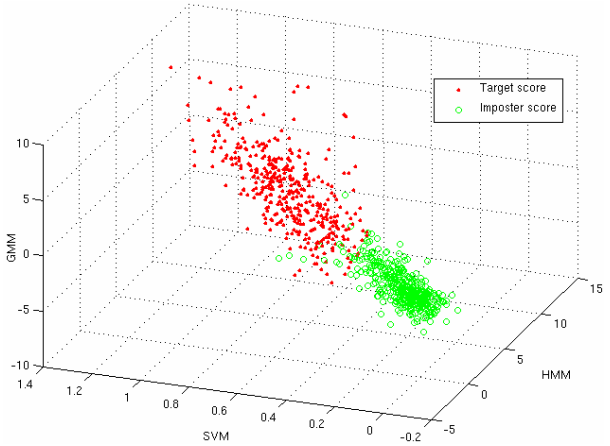

Figure 6: *3-D distribution of the scores for target and imposter trials (HMM, GMM and SVM scores)*

6. Conclusions

HMM supervectors as features in SVM based text-dependent speaker verification are investigated in this paper. By adapting the mixture components to all states within HMMs, an HMM supervector is constructed. We have evaluated two kernels for SVMs with HMM supervectors. The normalized SVM output scores can achieve additional performance gains.

It was shown that SVM can be very effective in improving the performance of existing HMM based verification methods. Experimental results show that fusion of HMM and SVM systems yield excellent results. The equal error rate was reduced from 4.01% to 3.47%. When the GMM system was incorporated to the fusion system, the performance will be further reduced to 2.95%.

7. References

[1] BenZeghiba, M. F., Bourlard, H., "Speaker verification based on user customized password", *IDIAP*, Martigny, Swizerland, 2001.

[2] Rosenberg, A. E., Lee, C. H., and Gokoen, S., "Connected word talker verification using whole word hidden Markov model", in *ICASSP-91*, vol. 1, pp. 381-384, 1991.

[3] Parthasarathy, S. and Rosenberg, A.E., "General phrase speaker verification using sub-word background models and likelihood-ratio scoring", *ICSLP*, vol. 4, pages 2403-2406, 1996.

[4] Li, Q., Juang, B. H, Zhou, Q., and Lee, C. H., "Automatic verbal information verification for user authentication", *IEEE Trans. ASSP*, vol. 8, no. 5, pp. 585-596, 2000.

[5] Campbell, W. M., "A SVM/HMM system for speaker recognition", in *ICASSP' 2003*, vol. 2, pp. 209-212, 2003.

[6] Wan, V. and Renals, S., "SVMSVM: support vector machine speaker verification methodology", *Proceeding of ICASSP 2003*, vol. 2, pp. 221-224, 2003.

[7] Fine, S., Navratil, J., Gopinath, R., "A hybrid GMM/SVM approach to speaker identification", in *Proc. of ICASSP'2001*, vol. 1, pp. 417-420, 2001.

[8] Campbell, W. M., Sturim, D. E., and Reynolds, D. A., "Support vector machines using GMM supervectors for speaker verification", *IEEE Signal Processing Letters*, vol. 13, no. 5, 2006.

[9] Campbell, W. M., Sturim, D. E., Reynolds, D. A., and Solomonoff, A., "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation", in *Proc. of ICASSP'2006*, 2006.

[10] Shimodaira, H., Noma, K., Nakai, M., Sagayama, S., "Support vector machine with dynamic time-alignment kernel for speech recognition", *Proc. Eurospeech 2001*. pp. 1841-1844. 2001.

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

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

[13] Solera-Urena, R., Martin-Iglesias, D., Gallardo-Antolin, A., Pelaez-Moreno, C. and Diaz-de-Maria, F., "Robust ASR using support vector machines", *Speech Communication*, vol. 49 (4), pp. 253-267, 2007.

[14] Auckenthaler, R., Carey, M., and Lloyd-thomas, H., "Score normalization for text-independent speaker verification systems", *IEEE on Digital Signal Processing*, vol. 10, pp. 42-54, 2000.

[15] Dong, C., Dong, Y., Huang, D., Guo, J., and Wang H., "A boosting approach for utterance verification", *Lecture Notes in Computer Science*, No. 4114, pp. 1170-1176, 2006.

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

[17] BenZeghiba, M. F. and H. Bourlard, "Hybrid HMM/ANN and GMM combination for user-customized password speaker verification", *Proceeding of ICASSP 2003*, vol. 2 pp. 225-228, 2003

[18] Ferrer, L., et al, "Modeling Duration Patterns for Speaker Recognition", *Proc. Eurospeech 2003*. pp. 2017-2020. 2003.