A MULTI-CLASS MLLR KERNEL FOR SVM SPEAKER RECOGNITION

Zahi N. Karam[1], William M. Campbell[2]

[1]MIT Lincoln Laboratory, Lexington, MA, USA and MIT DSPG, Cambridge, MA, USA
[2]MIT Lincoln Laboratory, Lexington, MA, USA
zahi@mit.edu, wcampbell@ll.mit.edu

ABSTRACT

Speaker recognition using support vector machines (SVMs) with features derived from generative models has been shown to perform well. Typically, a universal background model (UBM) is adapted to each utterance yielding a set of features that are used in an SVM. We consider the case where the UBM is a Gaussian mixture model (GMM), and maximum likelihood linear regression (MLLR) adaptation is used to adapt the means of the UBM. Recent work has examined this setup for the case where a global MLLR transform is applied to all the mixture components of the GMM UBM. This work produced positive results that warrant examining this setup with multi-class MLLR adaptation, which groups the UBM mixture components into classes and applies a different transform to each class. This paper extends the MLLR/GMM framework to the multi-class case. Experiments on the NIST SRE 2006 corpus show that multi-class MLLR improves on global MLLR and that the proposed system's performance is comparable with state of the art systems.

Index Terms— Speaker recognition, MLLR, Support vector machine, Kernel, Adaptation

1. INTRODUCTION

SVMs have become a popular and powerful tool in text-independent speaker verification. At the core of any SVM system is a choice of SVM feature expansion and an associated choice of kernel. The feature expansion maps a given utterance to a feature vector in a high-dimensional SVM feature space, and the kernel induces a distance metric in this space. A recent trend has been to derive expansions by adapting a UBM to an utterance-specific model.

Recent work [1] used MLLR to adapt the means of a GMM UBM to a given utterance, and the kernel used was the Gaussian supervector (GSV) kernel. The GSV kernel [2] is derived from an approximation of the KL divergence between two adapted GMMs and corresponds to a weighted inner product between the Gaussian supervectors (GSVs), which are vectors formed by stacking the means of the adapted GMMs. The work in [1] focused on MLLR adaptation that applied the same global affine transformation to the means of all the mixture components of the UBM. It also presented an equivalent implementation of the GSV kernel, called the MLLRSV kernel, as a weighted inner product between the MLLR transform-vectors, which are formed by stacking the elements of the affine transformation. The results of this work were promising and warranted further exploration.

* 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. This work was also supported by MIT Lincoln Laboratory PO 3077828.

In this paper, we expand on this recent work by allowing for multiple classes in the MLLR adaptation: this groups the mixture components of the GMM UBM into classes and applies a different transform to each of the classes. We introduce the extension of the GSV kernel and the MLLRSV kernel for the multi-class MLLR case. We also present the details of the implementation of this extended system. Multi-class MLLR is also used in the LVCSR/SVM system proposed in [3] where it is used to adapt a large vocabulary speech recognition system (LVCSR). We therefore briefly discuss how the MLLRSV kernel can be used in conjunction with an LVCSR system as we believe it will improve the performance over the currently used kernels. It is important to note that one of the goals of this work is to demonstrate that a simpler GMM UBM designed specifically for speaker recognition can be used in place of an LVCSR system to obtain gains with multiple classes. Throughout this paper we will present in detail the implementation of the two class case, extension to a larger number of classes is straightforward. We also present and discuss results for the two and four-class cases.

2. BACKGROUND

2.1. Support Vector Machines

An SVM [4] is a two-class classifier constructed from sums of a kernel function $K(\cdot,\cdot)$,

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

where the t_i are the ideal outputs, $\sum_{i=1}^{L} \gamma_i t_i = 0$, and $\gamma_i > 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization process [5]. 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.

2.2. Maximum Likelihood Linear Regression

Maximum likelihood linear regression (MLLR) adaptation adapts the means of the mixture components of a GMM by applying an affine transformation. The same affine transform may be shared by all the mixture components:

$$\mathbf{m}_i = \mathbf{A}\bar{\mathbf{m}}_i + \mathbf{b} \qquad \forall i, \tag{2}$$

where $\bar{\mathbf{m}}_i$ are the means of the unadapted GMM, and $\mathbf{m}_i$ are the adapted means.

Alternatively, the mixture components may be grouped into classes and a different affine transform shared by all the mixture components in each of the classes:

$$\mathbf{m}_i = \mathbf{A}_1\bar{\mathbf{m}}_i + \mathbf{b}_1 \qquad \forall \mathbf{m}_i \in \text{Class}_1, \tag{3}$$

$$\mathbf{m}_i = \mathbf{A}_2\bar{\mathbf{m}}_i + \mathbf{b}_2 \qquad \forall \mathbf{m}_i \in \text{Class}_2. \tag{4}$$

In both the single and multi-class cases the transforms are chosen to maximize the likelihood that the utterance was generated by the adapted model [6]. The MLLR algorithm computes the transforms $\mathbf{A}$ and $\mathbf{b}$, not the transformed means $\mathbf{m}_i$ and subsequently additional computation is needed to obtain the transformed means.

Multi-class MLLR adaptation allows for more freedom in adapting the GMM, since all the means are not constrained to move the same way. The choice of how to group mixture components into the different classes and the number of classes is non-trivial. One can group the mixture components via a data-driven approach that combines together mixture components that are close in acoustic space. Alternatively, as in this paper, the grouping can be done based on broad phonetic classes. We explore the two and four-class cases: the two-class case groups sonorants into one class and obstruents into the other, the four-class case further divides the sonorants into vowels and sonorant consonants and the obstruents into fricatives and stops. The two and four-class break-up is presented in Figure 1.

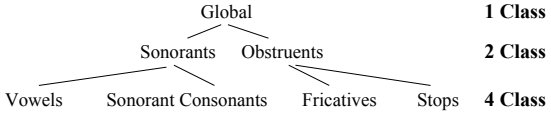

Fig. 1. *Class-division tree structure.*

As the number of classes increases the amount of adaptation data assigned to each class decreases. This leads to instances where there is not enough adaptation data to obtain a good transform for a given class. A common method to handle these instances is to "back-off" from the class-specific transform and use a more general one to transform the means of that class. For example, if there is not enough data to obtain a transform for the vowels we back-off and use the transform for the sonorants to adapt the vowels. More details on how the mixture components were chosen and the back-off technique used will follow in Section 5.

3. MLLR FEATURE EXPANSIONS

The SVM feature expansion is a map between an utterance and a high-dimensional vector in the SVM feature space. We will focus on the case of two-class MLLR adaptation and will present two expansions which are byproducts of this adaptation.

The UBM, used to model a wide range of speakers, is an N mixture diagonal covariance GMM, $g(\mathbf{x})$. It is formed by a weighted sum of two $N/2$ mixture GMMs: the first $N/2$ mixture components are assigned to the sonorants and the rest to the obstruents. The process of assigning components and the choice of the weighting ($\mu_\mathbf{s}$ and $\mu_\mathbf{o}$) are discussed in more detail in Section 5.

$$g(\mathbf{x}) = \mu_\mathbf{s} \sum_{i=1}^{N/2} \lambda_i \mathcal{N}(\mathbf{x}; \bar{\mathbf{m}}_i, \mathbf{\Sigma}_i) + \mu_\mathbf{o} \sum_{i=N/2+1}^{N} \lambda_i \mathcal{N}(\mathbf{x}; \bar{\mathbf{m}}_i, \mathbf{\Sigma}_i),$$

where $\mathcal{N}(\mathbf{x}; \bar{\mathbf{m}}_i, \mathbf{\Sigma}_i)$ is a Gaussian with mean $\bar{\mathbf{m}}_i$ and covariance $\mathbf{\Sigma}_i$. Adapting the means of the UBM via two-class MLLR to a given utterance utt_α produces a transformation matrix $\mathbf{A_s}$ and offset vector $\mathbf{b_s}$ for the sonorants, which can be used to adapt the means of the UBM assigned to the sonorants, and $\mathbf{A_o}$ and $\mathbf{b_o}$ for the obstruents, which can be used to adapt the means of the UBM assigned to the obstruents.

The first expansion is the Gaussian supervector $\mathbf{m}$, which is constructed by stacking the means of the adapted model. The second is the MLLR transform-vector $\boldsymbol{\tau}$ which consists of stacking the transposed rows of the transform matrix $\mathbf{A_s}$ separated by the corresponding entries of the vector $\mathbf{b_s}$ followed by the transposed rows of $\mathbf{A_o}$ separated by the corresponding entries of $\mathbf{b_o}$. The process is shown in Figure 2.

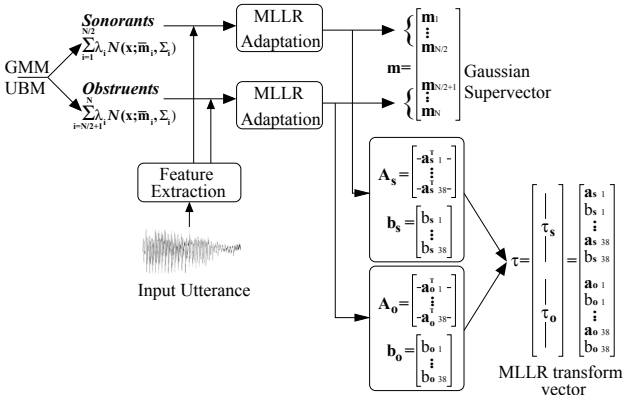

Fig. 2. *Two choices of feature expansions for the two-class case.*

4. MLLR KERNELS

A major component of an SVM system is the kernel, which defines a distance between two different points in the SVM feature space. In our context, this translates to defining a distance between two utterances. In this section, we will discuss the Gaussian supervector (GSV) kernel we have used and an *equivalent* implementation in the MLLR transform feature space, which we call the MLLRSV kernel. Our focus on the GSV kernel is motivated by [1] which compared it to other kernels and showed that it outperformed them.

4.1. Gaussian Supervector (GSV) Kernel

Suppose we have two utterances, utt_α and utt_β. We adapt the GMM UBM $g(\mathbf{x})$, via MLLR adaptation of the means, to obtain two new GMMs, $g_\alpha(\mathbf{x})$ and $g_\beta(\mathbf{x})$ respectively, that represent the utterances. The GSV kernel, $K_{SV}(\text{utt}_\alpha, \text{utt}_\beta)$, is derived in [2] by upperbounding the KL divergence between the two new GMMs:

$$K_{SV}(\text{utt}_\alpha, \text{utt}_\beta) = \sum_{i=1}^{N} \left(\sqrt{\lambda_i}\mathbf{\Sigma}_i^{-\frac{1}{2}}\mathbf{m}_i^\alpha\right)^t \left(\sqrt{\lambda_i}\mathbf{\Sigma}_i^{-\frac{1}{2}}\mathbf{m}_i^\beta\right) \quad (5)$$

$$= \mathbf{m}^{\alpha t}\mathbf{\Delta}\mathbf{m}^\beta, \quad (6)$$

where $\mathbf{m}^\alpha$ and $\mathbf{m}^\beta$ are the Gaussian supervectors of the utterances and $\mathbf{\Delta} = diag(\sqrt{\lambda_1}\mathbf{\Sigma}_1^{-1}, ..., \sqrt{\lambda_N}\mathbf{\Sigma}_N^{-1})$ is a diagonal matrix since the $\mathbf{\Sigma}_i$s are also diagonal matrices.

Since $\mathbf{\Delta}$ is a diagonal matrix and $\mathbf{m}$ is the stacked means of the different classes, then the multi-class extension to the GSV kernel is:

$$K_{SV}(\text{utt}_\alpha, \text{utt}_\beta) = \mu_\mathbf{s} K_{SV,S}(\text{utt}_\alpha, \text{utt}_\beta) + \mu_\mathbf{o} K_{SV,O}(\text{utt}_\alpha, \text{utt}_\beta), \quad (7)$$

where $K_{SV,S}(\text{utt}_\alpha, \text{utt}_\beta)$ and $K_{SV,O}(\text{utt}_\alpha, \text{utt}_\beta)$ are the class-dependent GSV kernels for the sonorants and obstruents respectively.

4.2. MLLRSV Kernel

Multi-class MLLR transforms the means of all the mixture components in a given class of the GMM UBM by the same affine transformation, as in equations (3) and (4). This constraint allows us to derive a MLLRSV kernel in MLLR transform-vector space that is *equivalent* to the GSV kernel. We begin by replacing the adapted means in equation (7) with the affine transforms of the UBM means.

$\mathbf{A_s}$, $\mathbf{A_o}$, $\mathbf{b_s}$, $\mathbf{b_o}$ are the transforms for utt_α and $\mathbf{C_s}$, $\mathbf{C_o}$, $\mathbf{d_s}$, $\mathbf{d_o}$ are the transforms for utt_β.

$$K_{SV}(\text{utt}_\alpha, \text{utt}_\beta) = \mu_\mathbf{s} K_{SV,S}(\text{utt}_\alpha, \text{utt}_\beta) + \mu_\mathbf{o} K_{SV,O}(\text{utt}_\alpha, \text{utt}_\beta)$$
$$= \mu_\mathbf{s} \sum_{i=1}^{N/2} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}(\mathbf{A_s}\bar{\mathbf{m}}_i + \mathbf{b_s})\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}(\mathbf{C_s}\bar{\mathbf{m}}_i + \mathbf{d_s})\right) +$$
$$\mu_\mathbf{o} \sum_{i=N/2+1}^{N} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}(\mathbf{A_o}\bar{\mathbf{m}}_i + \mathbf{b_o})\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}(\mathbf{C_o}\bar{\mathbf{m}}_i + \mathbf{d_o})\right), \quad (8)$$

where $\bar{\mathbf{m}}_i$ is the mean vector of the i^{th} mixture component of the UBM, the diagonal matrix $\boldsymbol{\Delta}_i = \lambda_i \boldsymbol{\Sigma}_i^{-1}$. Expanding the sonorant part of equation (8) yields

$$\begin{aligned} K_{SV,S}(\text{utt}_\alpha, \text{utt}_\beta) = & \sum_{i=1}^{N/2} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{b_s}\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{d_s}\right) \\ + & \sum_{i=1}^{N/2} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{A_s}\bar{\mathbf{m}}_i\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{C_s}\bar{\mathbf{m}}_i\right) \\ + & \sum_{i=1}^{N/2} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{A_s}\bar{\mathbf{m}}_i\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{d_s}\right) \\ + & \sum_{i=1}^{N/2} \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{b_s}\right)^t \left(\boldsymbol{\Delta}_i^{\frac{1}{2}}\mathbf{C_s}\bar{\mathbf{m}}_i\right). \quad (9) \end{aligned}$$

After some manipulation, details of which are in [1], we obtain:

$$\begin{aligned} K_{SV,S}(\text{utt}_\alpha, \text{utt}_\beta) = & \sum_{k=1}^{M} b_{\mathbf{s}k} d_{\mathbf{s}k} \delta_{\mathbf{s}k} + \sum_{k=1}^{M} \mathbf{a}_{\mathbf{s}k}^t \mathbf{R}_{\mathbf{s}k} \mathbf{c}_{\mathbf{s}k} \\ + & \sum_{k=1}^{M} d_{\mathbf{s}k} \mathbf{a}_{\mathbf{s}k}^t \mathbf{r}_{\mathbf{s}k} + \sum_{k=1}^{M} b_{\mathbf{s}k} \mathbf{r}_{\mathbf{s}k}^t \mathbf{c}_{\mathbf{s}k} \quad (10) \\ = & \boldsymbol{\tau}_{\mathbf{s}\alpha}^t \mathbf{Q_s} \boldsymbol{\tau}_{\mathbf{s}\beta} \quad (11) \end{aligned}$$

where M is the number of rows in $\mathbf{A_s}$, $\mathbf{a}_{\mathbf{s}k}$ and $\mathbf{c}_{\mathbf{s}k}$ are the transpose of the k^{th} rows of $\mathbf{A_s}$ and $\mathbf{C_s}$ respectively, $b_{\mathbf{s}k}$ and $d_{\mathbf{s}k}$ are the k^{th} elements of $\mathbf{b_s}$ and $\mathbf{d_s}$ respectively, Δ_{ik} is the k^{th} diagonal element of the diagonal matrix $\boldsymbol{\Delta}_i$, $\mathbf{R}_{\mathbf{s}k} = \sum_{i=1}^{N/2} \Delta_{ik} \bar{\mathbf{m}}_i \bar{\mathbf{m}}_i^t$, $\mathbf{r}_{\mathbf{s}k} = \sum_{i=1}^{N/2} \Delta_{ik} \bar{\mathbf{m}}_i$, $\delta_{\mathbf{s}k} = \sum_{i=1}^{N/2} \Delta_{ik}$, $\boldsymbol{\tau}_{\mathbf{s}\alpha}$ and $\boldsymbol{\tau}_{\mathbf{s}\beta}$ are the sonorant parts of the MLLR transform-vectors of the utterances, and $\mathbf{Q_s}$ is a block diagonal matrix consisting of M blocks $\mathbf{Q}_{\mathbf{s}k}$ of size $(M+1)$x$(M+1)$. Equation (12) shows the structure of the blocks:

$$\mathbf{Q}_{\mathbf{s}k} = \begin{pmatrix} \mathbf{R}_{\mathbf{s}k} & \mathbf{r}_{\mathbf{s}k} \\ \mathbf{r}_{\mathbf{s}k}^t & \delta_{\mathbf{s}k} \end{pmatrix}. \quad (12)$$

Note that the summations in $\mathbf{R}_{\mathbf{s}k}$, $\mathbf{r}_{\mathbf{s}k}$ and $\delta_{\mathbf{s}k}$ are from $i = 1$ to $N/2$, only over the mixture components pertaining to the sonorant class. With this in mind the form of the obstruent part of the kernel is

$$K_{SV,O}(\text{utt}_\alpha, \text{utt}_\beta) = \boldsymbol{\tau}_{\mathbf{o}\alpha}^t \mathbf{Q_o} \boldsymbol{\tau}_{\mathbf{o}\beta}, \quad (13)$$

where the summations in $\mathbf{R}_{\mathbf{o}k}$, $\mathbf{r}_{\mathbf{o}k}$ and $\delta_{\mathbf{o}k}$ are from $i = N/2 + 1$ to N, only over the mixture components pertaining to the obstruent class.

From equations (11) and (13) we note that the GSV kernel can be written as a weighted inner product between the MLLR transform-vectors.

$$\begin{aligned} K_{SV}(\text{utt}_\alpha, \text{utt}_\beta) = & \begin{bmatrix} \boldsymbol{\tau}_{\mathbf{s}\alpha}^t & \boldsymbol{\tau}_{\mathbf{o}\alpha}^t \end{bmatrix} \begin{bmatrix} \mu_\mathbf{s}\mathbf{Q_s} & \mathbf{0} \\ \mathbf{0} & \mu_\mathbf{o}\mathbf{Q_o} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_{\mathbf{s}\beta} \\ \boldsymbol{\tau}_{\mathbf{o}\beta} \end{bmatrix} \\ = & \boldsymbol{\tau}_\alpha^t \mathbf{Q} \boldsymbol{\tau}_\beta \quad (14) \end{aligned}$$

It is important to note that since the $\mathbf{Q}$ matrix depends only on the UBM means, covariances and mixture weights it can be computed offline.

An advantage of equation (14) over equation (7) is that the number of multiplies it requires only depends on the size of the GMM feature vectors and number of MLLR classes not on the number of mixture components in the GMM. Another advantage is that it does not require transforming the means, this saves computation and removes the need for storing the adapted means. These two advantages and the block diagonal structure of $\mathbf{Q}$ result in and order of magnitude reduction in multiplies, for our system (details in [1]), by using the MLLRSV implementation over the original GSV kernel; this reduction becomes more significant as the number of mixture components increases.

4.3. MLLRSV Kernel for LVCSR systems

The LVCSR/SVM system presented in [3] uses MLLR adaptation with a speaker independent LVCSR system and a kernel consisting of an inner product between rank-normalized transform-vectors. Recent results, presented in [1], showed the advantage of the GSV kernel over other kernels that are inner products between normalized MLLR transform-vectors, including the one used in [3], for the case where the UBM is a GMM. Unfortunately, the GSV kernel, if applied in its original form (7), can be computationally prohibitive since the number of multiplies increases as $O(N^2)$ where N is the number of Gaussian mixture components in the system, which is typically more than a hundred thousand for an LVCSR system. However, since MLLR adaptation is being used to adapt the means, one can follow the steps taken in Section 4.2 to derive a similar way to compute the GSV kernel in terms of an inner product between the MLLR transform-vectors significantly reducing computation.

5. IMPLEMENTATION

There are a number of issues that have to be addressed when building the multi-class MLLR/GMM system. The first, is how to divide the mixture components of the GMM into multiple classes. For the two-class case, we chose to perform the divide along broad phonetic classes: sonorants and obstruents. However, since our UBM is not an LVCSR system where it is clear which mixture components belong to which phoneme and thus to which of our two classes, we have to explicitly assign them: we assign the first $N/2$ mixture components to the sonorants class and the remaining $N/2$ to the obstruents class. We also perform open-loop phonetic recognition on all the data used to train the UBM, the background, and the speaker recognition system and to test the system; this allows us to assign which part of the data will be used to train/test each class. We also tried unequal splitting of the GMM mixture components amongst the classes, however this reduced performance.

Second, we use EM to train two class-dependent $N/2$ mixture GMMs each using the corresponding class-specific UBM training data. The N mixture GMM UBM is then created by combining the two $N/2$ mixture GMMs and scaling their weights so that the weights of the UBM add up to 1. The scaling, $\mu_\mathbf{s}$ and $\mu_\mathbf{o}$, is done according to the class priors, calculated as the percentage of frames assigned to each of the two classes in the background training data.

Third, the MLLR transformation matrix and offset vector for each of the two classes are computed by separately adapting, via MLLR, the class-dependent GMMs using only the frames of the adaptation utterance corresponding to each class. If the number of frames of the utterance assigned to a class is below a set number, empirically we chose 500, we back-off and use the full N mixture GMM and all the frames of the utterance to obtain the MLLR transformation matrix and vector. This transform computed by backing-off is then used to adapt *only* the $N/2$ means of the *original* class-dependent GMM. Similarly, in the four-class case if the number of

frames allocated to one of the four classes is below 250 then for that class one would back-off one level, e.g. from Vowels to Sonorants; if after backing-off one level the number of allocated frames is less than 500 then one would back-off one more level.

6. EXPERIMENTS

We performed experiments on the 2006 NIST speaker recognition (SRE) corpus. We focused on the single-side 1 conversation train, single-side 1 conversation test, and the multi-language handheld telephone task (the core test condition) [7]. This setup resulted in $3,612$ true trials and $47,836$ 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 and mean and variance normalization are applied to the features.

For the two-class case, two class-specific $N/2 = 256$ mixture GMM UBMs were trained using EM on the corresponding class-dependent data from the following corpora: Switchboard 2 phase 1, Switchboard 2 phase 4 (cellular), and OGI national cellular. These GMMs were combined with weights $\mu_{\mathbf{s}} = .71$ and $\mu_{\mathbf{o}} = .29$ to form a $N = 512$ mixture GMM UBM. For the four-class case, four class-specific $N/4 = 128$ mixture GMM UBMs were trained and combined to form a 512 mixture GMM with weights: .46 for vowels, .25 for sonorant consonants, .15 for fricatives, and .14 for stops.

We produced the SVM feature expansion on a per conversation (utterance) basis using multi-class MLLR adaptation. The adaptation was done per class-specific GMM. We used the HTK toolbox version 3.3 [8] to perform one iteration of MLLR to obtain the transformation. The various kernels were implemented using SVMTorch as an SVM trainer [5]. A background for SVM training consists of SVM features labeled as -1 extracted from utterances from example impostors [2]. An SVM background was obtained by extracting SVM features from 4174 conversations in a multi-language subset of the LDC Fisher corpus. In our experiments the size of the SVM features are $38 * 512 + 1$ for the supervector features and $38 * 39 + 1$ for the MLLR transform-vector features; note that we stack an element of value 1 at the end of each feature vector to incorporate the bias ξ into the SVM features.

For enrollment of target speakers, we produced 1 SVM feature vector per conversation side. We then trained an SVM model using the target SVM feature and the SVM background. This resulted in selecting support vectors from the target speaker and background SVM feature vectors and assigning the associated weights.

7. RESULTS AND DISCUSSION

We compared the global single class MLLR/GMM and GSV kernel system (1C_MLLRSV) with the two and four-class MLLR/GMM and GSV kernel systems (2C_MLLRSV and 4C_MLLRSV) and a state of the art MAP/GMM system (MAPSV) described in [2] where the *same* GMM UBM is adapted via MAP adaptation and the GSV kernel is used. Equal error rates (EER) and NIST minimum decision cost functions (DCF) for the various kernels are shown in Table 1.

Table 1. EER and min DCF scores.

Kernel	EER	min DCF
1C_MLLRSV	9.46%	.039
2C_MLLRSV	7.81%	.035
4C_MLLRSV	8.19%	.037
MAPSV	7.24%	.031

In [1] we showed that the GSV kernel could be computed efficiently as an inner product between MLLR-transform vectors if MLLR is used to adapt the means of the GMM UBM, and that the GSV kernel outperformed other kernels that are inner products between MLLR-transform vectors. We had speculated that using multi-class MLLR, as in the LVCSR/SVM system presented in [3] which uses eight-class MLLR adaptation, would improve performance. As expected, the two-class system yields a 15% improvement over the single class system, however there was no further improvement for the four-class system. This lack of improvement for the four-class is most likely due to the unstable transcripts provided by the open-loop phonetic recognizer, which become less reliable as the number of classes increases. It is important to note that the gain in performance cause by two-class MLLR does require additional computation due to the phonetic recognition.

In this paper we have focused on attempting to understand the improvement that multi-class MLLR could provide over the single class, and accordingly we kept the total number of mixtures in the GMM UBM to $N = 512$. Thus, even though there are similarities between our system and the LVCSR/SVM system, they cannot be directly compared because of the miss-match between the total number of mixture components in the UBM and the stability of the transcripts provided by the systems. Specifically, the LVCSR system has more than a hundred thousand mixture components while we use 512 and the transcripts provided by the LVCSR are significantly more stable than the ones provided by our open-loop phonetic recognizer. An avenue of future work is to explore these issues further.

8. CONCLUSION

This paper expands the MLLR/GMM SVM speaker recognition framework presented in [1] to handle multi-class MLLR adaptation. This extension allows for greater flexibility in adapting the means of the GMM UBM resulting, for the two-class case, in a 15% relative improvement in performance over the single class case, which approaches that of the state of the art MAP/GMM SVM system [2]. However, further increasing the number of classes does not yield better results as is seen from the four-class scores. The paper also derives the implementation of the MLLRSV for the multi-class case, which is a computationally efficient implementation of the Gaussian supervector kernel that scales linearly with the number of transforms and is independent of the number of mixture components.

9. REFERENCES

[1] Zahi N. Karam and William M. Campbell, "A new kernel for SVM MLLR based speaker recognition," in *Proc. Interspeech*, 2007.

[2] 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.

[3] A. Stolcke, L. Ferrer, and S. Kajarekar, "Improvements in MLLR-transform-based speaker recognition," in *Proc. Odyssey06*, 2006, pp. 1–6.

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

[5] 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.

[6] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Language*, vol. 10, no. 4, pp. 249–264, 1996.

[7] "The NIST year 2006 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/2006/index.htm, 2006.

[8] S. Young et al, "The HTK book," http://htk.eng.cam.ac.uk.