A New Kernel for SVM MLLR based 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. We examine two possible SVM feature expansions that arise in this context: the first, a GMM supervector is constructed by stacking the means of the adapted GMM, and the second consists of the elements of the MLLR transform. We examine several kernels associated with these expansions. We show that both expansions are equivalent given an appropriate choice of kernels. Experiments performed on the NIST SRE 2006 corpus clearly highlight that our choice of kernels, which are motivated by distance metrics between GMMs, outperform ad-hoc ones. We also apply SVM nuisance attribute projection (NAP) to the kernels as a form of channel compensation and show that, with a proper choice of kernel, we achieve results comparable to existing SVM based recognizers.

Index Terms: speaker recognition, MLLR, SVM, supervector

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 from adapting a UBM to an utterance-specific model. A system proposed in [1] uses MLLR to adapt, via an affine transformation, the means of a speaker independent automatic speech recognition (ASR) system to a given utterance. The entries of the affine transforms are then used as the feature vectors. Another system uses constrained MLLR (CMLLR) to adapt the means and covariances of a GMM UBM to a given utterance and uses the entries of the transform as features [2]. Maximum a-posteriori (MAP) adaptation is used in [3] to adapt the means of a GMM UBM, and the corresponding feature vector is the Gaussian supervector (GSV) which consists of the stacked adapted means.

We have chosen to explore the use of MLLR to adapt the means of a GMM UBM, similar to [2]. In this context, we use two SVM feature expansions. The first is the GSV, and the second uses entries of the affine transform computed by MLLR. The choice for using MLLR and a GMM UBM is motivated by several reasons. Most notably it does not require the overhead of performing ASR, and the constrained nature of the MLLR transform may help mitigate channel effects. Note that although we restrict ourselves to GMM adaptation, our methods are applicable to the general ASR case.

The breakdown of this paper is as follows. Section 1 gives a brief overview of SVMs, MLLR adaptation, and NAP channel compensation. Section 2 presents the two choices of MLLR feature expansions. Section 3 explores several choices of kernels in this context. Section 4 contains the details of the performed experiments. Section 5 presents our results and suggests future work. Finally, we conclude in Section 6.

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.

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

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

where $\phi(\mathbf{x})$ is a mapping from the input space (where $\mathbf{x}$ lives) to a possibly infinite-dimensional SVM feature space. We will refer to the $\phi(\mathbf{x})$ as the SVM features.

2.2. Maximum Likelihood Linear Regression

In maximum likelihood linear regression (MLLR), a *single* affine transformation is applied to the means of *all* the mixtures of the GMM UBM to obtain the adapted means:

$$\mathbf{m}_i = \mathbf{A}\bar{\mathbf{m}}_i + \mathbf{b} \qquad \forall i, \tag{3}$$

where $\mathbf{A}$ and $\mathbf{b}$ are chosen to maximize the likelihood that the utterance was generated by the adapted model [6]. In equation (3), $\bar{\mathbf{m}}_i$ are the means of the UBM and $\mathbf{m}_i$ are the adapted

* 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.

means. It is important to note that the MLLR algorithm computes $\mathbf{A}$ and $\mathbf{b}$, not the transformed means $\mathbf{m}_i$ and subsequently the additional computation suggested by equation (3) is used to obtain the transformed means.

2.3. Nuisance Attribute Projection

Nuisance Attribute Projection (NAP) is a method for improving performance in SVM speaker recognition with knowledge of nuisance attributes such as channel, session, language, etc. [3]. Our emphasis in this paper will be on session variation. The goal of NAP is to find a low corank projection operator $\mathbf{P}$ such that the new kernel

$$K_{\mathrm{NAP}}(\mathbf{x},\mathbf{y}) = (\mathbf{P}\phi(\mathbf{x}))^t(\mathbf{P}\phi(\mathbf{y})) \tag{4}$$

is not sensitive to channel effects. Note that NAP is applied to the SVM feature space after deriving an appropriate feature expansion.

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 two expansions which are byproducts of MLLR adaptation.

The UBM we use is an $N = 512$ mixture diagonal covariance GMM , $g(\mathbf{x})$, that models a wide range of speakers,

$$g(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(\mathbf{x}; \bar{\mathbf{m}}_i, \mathbf{\Sigma}_i), \tag{5}$$

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 MLLR to a given utterance utt_α produces a transformation matrix $\mathbf{A}$ and offset vector $\mathbf{b}$ which can be used to compute a new set of means $\mathbf{m}_i^\alpha$. The first expansion is the GMM 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 $\mathbf{A}$ separated by the corresponding entries of the vector $\mathbf{b}$. The process is shown in Figure 1.

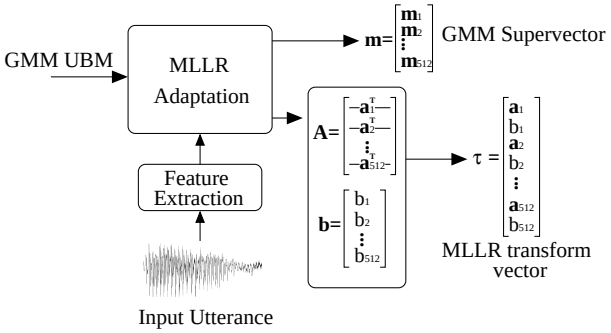

Figure 1: *Two choices of feature expansions.*

4. Choice of Kernels

A major component of an SVM 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 different kernels we have explored. Our focus will be on the supervector kernel since it is well-motivated and produces the best results.

4.1. MLLR Supervector 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. This results in GMM supervectors, $\mathbf{m}^\alpha$ and $\mathbf{m}^\beta$. A natural distance between the two utterances is the KL divergence between the two adapted GMMs,

$$D(g_\alpha \| g_\beta) = \int_{R^n} g_\alpha(\mathbf{x}) \log\left(\frac{g_\alpha(\mathbf{x})}{g_\beta(\mathbf{x})}\right) d\mathbf{x} \tag{6}$$

Unfortunately, the KL divergence does not satisfy the Mercer condition, so using it in an SVM is difficult.

Instead of using the KL divergence directly, we consider an approximation [3] which upper bounds it,

$$d(\mathbf{m}^\alpha, \mathbf{m}^\beta) = \frac{1}{2}\sum_{i=1}^{N} \lambda_i (\mathbf{m}_i^\alpha - \mathbf{m}_i^\beta)\mathbf{\Sigma}_i^{-1}(\mathbf{m}_i^\alpha - \mathbf{m}_i^\beta). \tag{7}$$

The distance in (7) has a corresponding kernel function [3],

$$K_{SV}(\mathrm{utt}_\alpha, \mathrm{utt}_\beta) = \sum_{i=1}^{N} \left(\sqrt{\lambda_i}\mathbf{\Sigma}_i^{-\frac{1}{2}}\mathbf{m}_i^\alpha\right)^t \left(\sqrt{\lambda_i}\mathbf{\Sigma}_i^{-\frac{1}{2}}\mathbf{m}_i^\beta\right). \tag{8}$$

The MLLR supervector (MLLRSV) 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.

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

$$f(\mathbf{x}) = \left(\sum_{i=1}^{L} \gamma_i t_i \phi(\mathbf{x}_i)\right)^t \phi(\mathbf{x}) + \xi = \mathbf{w}^t\phi(\mathbf{x}) + \xi. \tag{9}$$

This means we only have to compute a single inner product between the target model and the GMM supervector in scoring.

4.2. MLLR Supervector Kernel in MLLR Transform Space

MLLR adaptation transforms the means of all the mixtures of the UBM GMM by the same affine transformation in equation (3). This constraint allows us to derive a nonlinear kernel in MLLR transform-vector space that is equivalent to the supervector kernel. We begin by replacing the adapted means in equation (8) with the affine transform of the UBM means.

$$K_{SV}(\mathrm{utt}_\alpha, \mathrm{utt}_\beta) = \sum_{i=1}^{N} \left(\mathbf{\Delta}_i^{\frac{1}{2}}(\mathbf{A}\bar{\mathbf{m}}_i + \mathbf{b})\right)^t \left(\mathbf{\Delta}_i^{\frac{1}{2}}(\mathbf{C}\bar{\mathbf{m}}_i + \mathbf{d})\right), \tag{10}$$

where N is the number of mixtures of the UBM, $\bar{\mathbf{m}}_i$ is the mean vector of the i^{th} mixture of the UBM, and $\mathbf{\Delta}_i = \lambda_i\mathbf{\Sigma}_i^{-1}$ which is diagonal. Expanding equation (10) yields

$$\begin{aligned} & K_{SV}(\mathrm{utt}_\alpha, \mathrm{utt}_\beta) = \\ = \;& \textstyle\sum_{i=1}^{N} \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{A}\bar{\mathbf{m}}_i\right)^t \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{C}\bar{\mathbf{m}}_i\right) \\ + \;& \textstyle\sum_{i=1}^{N} \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{A}\bar{\mathbf{m}}_i\right)^t \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{d}\right) \\ + \;& \textstyle\sum_{i=1}^{N} \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{b}\right)^t \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{C}\bar{\mathbf{m}}_i\right) \\ + \;& \textstyle\sum_{i=1}^{N} \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{b}\right)^t \left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{d}\right). \end{aligned} \tag{11}$$

We will focus on the first term in equation (11). Note that tr($\mathbf{A}$) is the trace of the matrix $\mathbf{A}$, $\mathbf{e}_k$ is a vector that has a value of 1 as its k^{th} entry and 0 for every other entry, Δ_{ik} is the k^{th} diagonal element of the diagonal matrix $\mathbf{\Delta}_i$, M is the number of rows in $\mathbf{A}$, and that $\mathbf{a}_k$ is the transpose of the k^{th} row of the matrix $\mathbf{A}$.

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{A}\bar{\mathbf{m}}_i\right)^t\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{C}\bar{\mathbf{m}}_i\right)\\
&= \sum_{i=1}^{N}\text{tr}\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{A}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t\mathbf{C}^t\mathbf{\Delta}_i^{\frac{1}{2}}\right)\\
&= \sum_{i=1}^{N}\text{tr}\left(\mathbf{\Delta}_i\mathbf{A}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t\mathbf{C}^t\right)\\
&= \sum_{i=1}^{N}\text{tr}\left((\sum_{k=1}^{M}\Delta_{ik}\mathbf{e}_k\mathbf{e}_k^t)\mathbf{A}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t\mathbf{C}^t\right)\\
&= \sum_{k=1}^{M}\sum_{i=1}^{N}\text{tr}\left(\mathbf{e}_k\mathbf{e}_k^t\mathbf{A}(\Delta_{ik}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t)\mathbf{C}^t\right)\\
&= \sum_{k=1}^{M}\text{tr}\left(\mathbf{e}_k^t\mathbf{A}(\sum_{i=1}^{N}\Delta_{ik}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t)\mathbf{C}^t\mathbf{e}_k\right)\\
&= \sum_{k=1}^{M}\mathbf{a}_k^t(\sum_{i=1}^{N}\Delta_{ik}\bar{\mathbf{m}}_i\bar{\mathbf{m}}_i^t)\mathbf{c}_k\\
&= \sum_{k=1}^{M}\mathbf{a}_k^t\mathbf{R}_k\mathbf{c}_k. \qquad (12)
\end{aligned}$$

In a similar fashion we can rewrite the remaining terms in equation (11) as follows:

$$\sum_{i=1}^{N}\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{A}\bar{\mathbf{m}}_i\right)^t\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{d}\right) = \sum_{k=1}^{M}d_k\mathbf{a}_k^t\mathbf{r}_k, \qquad (13)$$

$$\sum_{i=1}^{N}\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{b}\right)^t\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{C}\bar{\mathbf{m}}_i\right) = \sum_{k=1}^{M}b_k\mathbf{r}_k^t\mathbf{c}_k, \qquad (14)$$

$$\sum_{i=1}^{N}\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{b}\right)^t\left(\mathbf{\Delta}_i^{\frac{1}{2}}\mathbf{d}\right) = \sum_{k=1}^{M}b_kd_k\delta_k, \qquad (15)$$

where $\mathbf{r}_k = \sum_{i=1}^{N}\Delta_{ik}\bar{\mathbf{m}}_i$, b_k is the k^{th} element of the vector $\mathbf{b}$, and $\delta_k = \sum_{i=1}^{N}\Delta_{ik}$. Therefore the supervector kernel can be rewritten as

$$\begin{aligned}
K_{SV}(\text{utt}_\alpha, \text{utt}_\beta) = &\sum_{k=1}^{M}\mathbf{a}_k^t\mathbf{R}_k\mathbf{c}_k + \sum_{k=1}^{M}d_k\mathbf{a}_k^t\mathbf{r}_k\\
+ &\sum_{k=1}^{M}b_k\mathbf{r}_k^t\mathbf{c}_k + \sum_{k=1}^{M}b_kd_k\delta_k \qquad (16)\\
= &\boldsymbol{\tau}_\alpha^t\mathbf{Q}\boldsymbol{\tau}_\beta, \qquad (17)
\end{aligned}$$

where $\boldsymbol{\tau}$ is the MLLR transform-vector defined in Section 2.2.

The matrix $\mathbf{Q}$ must be positive-definite because equation (17) computes the same quantity as equation (8). $\mathbf{Q}$ is a block diagonal matrix consisting of M blocks $\mathbf{B}_k$ of size $(M+1)$x$(M+1)$. Equation (18) shows the structure of the blocks $\mathbf{B}_k$,

$$\mathbf{B}_k = \begin{pmatrix}\mathbf{R}_k & \mathbf{r}_k\\ \mathbf{r}_k^t & \delta_k\end{pmatrix}. \qquad (18)$$

It is important to note that since $\mathbf{Q}$ depends only on the UBM means, covariances and mixture weights it can be computed offline. The block-diagonal nature of $\mathbf{Q}$ also allows us to easily compute its square root. This in turn allows us to apply the model compaction technique in equation (9).

An advantage of equation (17) over equation (8) is that the number of multiplies it requires only depends on the size of the GMM feature vectors (38 in our case) and not on the number of mixtures in the GMM. Another advantage is that it does not require transforming the means which saves computation and removes the need for storing the adapted means. These two advantages and the block diagonal structure of $\mathbf{Q}$ result in an overall reduction of the number of multiplies from $O(M*(N+N^2))$ in equation (8) to $O((M+1)^3)$ in equation (17), where M is the size of the GMM feature vectors and N is the number of mixtures in the GMM. This equates to an order of magnitude reduction in the number of multiplies for our case. Note that this reduction in number of multiplies and storage requirements will have a significantly greater impact if this kernel is applied to an ASR speaker recognition system.

4.3. Alternative Kernels in MLLR Transform Space

We also explore four alternative kernels in MLLR transform-vectors. The first replaces the matrix $\mathbf{Q}$ by its diagonal approximation, which we refer to as the diag-supervector (DIAGSV) kernel. The second is the zero-one (Z-O) kernel which subtracts the means and divides by the standard deviations along each of the feature dimensions of the MLLR transform-vectors. The third is the Frobenius (FROB) kernel which does not apply any scale or shift to the MLLR transform-vectors; tr($[\mathbf{Ab}]^t[\mathbf{Cd}]$). The last is the rank-normalized (R-N) kernel which rank normalizes the MLLR transform-vectors.

5. 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.

The GMM UBM consists of 512 mixture components. 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 the SVM feature expansion on a per conversation (utterance) basis using MLLR adaptation. 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 [3]. 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 so we can incorporate the bias ξ into the SVM features.

For enrollment of target speakers, we produced 1 SVM feature from a single conversation. 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.

For SVM NAP, the desired corank was estimated on the 2005 NIST SRE corpus and the projection was designed based on session variability, see Section 2.3. The projection was

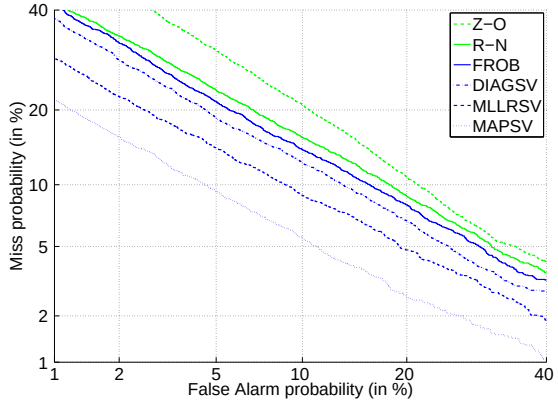

Figure 2: *Results for the various kernels without NAP.*

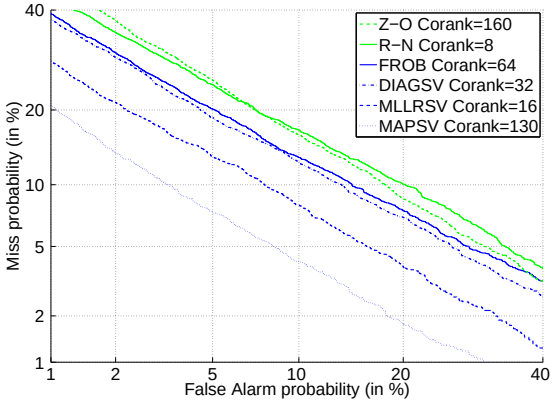

Figure 3: *Results for the various kernels with NAP.*

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 $\mathbf{P}^2 = \mathbf{P}$.

We compared the MLLRSV kernel, the DIAGSV kernel, the Z-O kernel, the FROB kernel, the R-N kernel, and a MAP supervector kernel (MAPSV) as in [3] where the UBM is adapted via MAP adaptation. Equal error rates (EER) and NIST minimum decision cost function (DCF) for the various kernels are shown in Table 1. DET curves for the experiments without and with NAP projection are plotted in Figures 2 and 3 respectively.

Table 1: EER and min DCF scores with and without NAP.

	No NAP		NAP		
Kernel	**EER**	**DCF**	**Corank**	**EER**	**DCF**
Z-O	14.95%	.064	160	13.23%	.056
R-N	13.19%	.051	8	13.61%	.052
FROB	12.38%	.05	64	11.45%	.048
DIAGSV	11.43%	.047	32	11.43%	.047
MLLRSV	9.46%	.039	16	8.92%	.038
MAPSV	7.24%	.031	130	6.29%	.030

6. Discussion and Future Work

The results show that of the examined kernels, the MLLRSV kernel yields the best performance, followed by the DIAGSV kernel. We believe the superiority of MLLRSV is due to its derivation from an approximation of the KL divergence as a distance between two GMMs. When examining the results for the linear kernels in MLLR transform-vector space we note that the diagonal approximation to the MLLRSV kernel produced the best results while the Z-O kernel produced the worst. To attempt and understand why the Z-O kernel performed poorly, we compared its scaling matrix to that of DIAGSV. The comparison showed that the Z-O kernel tended to emphasize dimensions that were weighted down by DIAGSV and vice versa. We also note that applying NAP tended to improve the EER by approximately 7.5% on average. Finally, we observe that the results of our MLLR supervector kernel are comparable to those of the MAPSV kernel.

A possible avenue for future work is to perform Y separate affine transformations on Y distinct subsets of mixtures of the GMM UBM as opposed to one global transformation for all the mixtures. It is straightforward to show that this will increase the sizes of the $\mathbf{Q}$ matrix and the $\boldsymbol{\tau}$ vector by a factor of Y. However, the structure of $\mathbf{Q}$ remains block diagonal with each block computed using the means, covariances and mixture weights of the corresponding subset. Our intuition indicates that exploring the use of multiple transforms may yield better performance than the MAPSV since we will be able to achieve a good balance between constraining the transformation and the dimensionality of the SVM feature space.

In a similar manner, the MLLR supervector kernel in transform-vector space in equation (17) can be applied to the case where the UBM is a speaker independent ASR system that is adapted via MLLR adaptation to the utterances, as in [1]. We expect that using the MLLR supervector kernel will improve performance of the system, similar to what we observed in our experiments.

7. Conclusion

This paper examined the use of SVMs, for speaker recognition, whose features are derived from adapting a GMM universal background model via maximum likelihood linear regression adaptation. The results clearly highlight the importance of choosing a properly motivated kernel. The main contribution of this paper is the formulation of the MLLR supervector kernel in MLLR transform-vector space. The advantage of this new formulation is that its storage and computation requirements do not increase with the number of mixtures. Experiments on the NIST SRE 2006 corpus showed the superiority of this kernel over ad-hoc kernels. Our work can be extended to the case where the UBM is a speaker independent ASR system to improve the performance of such a system.

8. Acknowledgments

The authors would like to thank Wade Shen for consultations on MLLR and an ASR SVM MLLR system. They would also like to thank Demba Ba for his contributions at the onset and his continued feedback.

9. References

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

[2] M. Ferras, C. Leung, C. Barras, and J. Gauvain, "Constrained MLLR for speaker recognition," in *ICASSP07 (to appear)*, 2007.

[3] 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.

[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.