Improving Eigenspace-based MLLR Adaptation by Kernel PCA

Brian Mak and *Roger Hsiao*

Department of Computer Science
Hong Kong University of Science & Technology, Hong Kong
{mak,hsiao}@cs.ust.hk

Abstract

Eigenspace-based MLLR (EMLLR) adaptation has been shown effective for fast speaker adaptation. It applies the basic idea of eigenvoice adaptation, and derives a small set of eigenmatrices using principal component analysis (PCA). The MLLR adaptation transformation of a new speaker is then a linear combination of the eigenmatrices. In this paper, we investigate the use of kernel PCA to find the eigenmatrices in the kernel-induced high dimensional feature space so as to exploit possible nonlinearity in the transformation supervector space. In addition, composite kernel is used to preserve the row information in the transformation supervector which, otherwise, will be lost during the mapping to the kernel-induced feature space. We call our new method *kernel eigenspace-based MLLR (KEMLLR) adaptation*. On a RM adaptation task, we find that KEMLLR adaptation may reduce the word error rate of a speaker-independent model by 11%, and outperforms MLLR and EMLLR adaptation.

1. Introduction

When the amount of adaptation speech is really small, say, a few seconds, eigenvoice-based adaptation methods [1, 2, 3, 4] have been shown more effective than the traditionally more popular methods such as the Bayesian-based MAP adaptation [5] and the transformation-based MLLR adaptation [6]. Eigenspace-based MLLR (EMLLR) adaptation [2] is a variant of the standard EV adaptation [1]. Instead of finding a small set of eigenvoices in the speaker supervector space as in the EV adaptation, EMLLR looks for a small set of eigenmatrices in the MLLR transformation supervector space. The acoustic model of a new speaker is then obtained by an MLLR transformation of the speaker-independent (SI) model, which is now a linear combination of the set of eigenmatrices.

Recently, we proposed an improvement to the EV adaptation called *kernel eigenvoice (KEV) adaptation* [7, 8] by exploiting possible nonlinearity in the speaker supervector space using kernel methods [9]. In this paper, we would like to apply similar kernel method to improve the performance of EMLLR adaptation. The basic idea is to map the speakers' MLLR transformation supervectors to a high dimensional feature space[1] via some nonlinear map, and then apply principal component analysis (PCA) there to derive the eigenmatrices in the *feature space*. During the actual computation, the exact nonlinear map need not be known, and the kernel eigenmatrices are obtained by *kernel PCA*. The computational procedure depends only on the inner products in the feature space, which can be obtained efficiently with a suitable kernel function. Our new method will be called *kernel eigenspace-based MLLR (KEMLLR) adaptation*.

One major challenge in KEMLLR adaptation is to preserve the row information in the transformation supervectors which, otherwise, will generally be lost during the mapping to the kernel-induced feature space. Our solution is the use of composite kernel.

2. Review of Eigenspace-based MLLR (EMLLR) Adaptation

Suppose there is a set of N speaker-dependent (SD) acoustic models which are hidden Markov models (HMMs) of the same topology with mixture Gaussian states. These SD models are estimated from the speaker-independent (SI) model by MLLR transformation. For simplicity, the following discussion assumes that only one global MLLR transform is used; its extension to multiple MLLR transforms using regression classes of Gaussians should be straight-forward. Thus, for the ith speaker, the mean vector of his gth Gaussian $\boldsymbol{\mu}_g^{(i)} \in \mathbb{R}^d$ is

$$\boldsymbol{\mu}_g^{(i)} = \mathbf{Y}^{(i)\prime}\boldsymbol{\xi}_g^{(si)}$$

where $\mathbf{Y}^{(i)\prime} \in \mathbb{R}^{d\times(d+1)}$ is the global MLLR transformation for the ith speaker, and $\boldsymbol{\xi}_g^{(si)} = [\boldsymbol{\mu}_g^{(si)\prime}, 1]'$ is the augmented mean vector of the corresponding Gaussian in the SI model. A *speaker transformation vector* is obtained by vectorizing $\mathbf{Y}$. (If we have multiple MLLR transformations, their vectorized matrices are stacked up to a *speaker transformation supervector*.) Let's denote $vec(\mathbf{Y})$ by $\mathbf{y}$. From the N transformation vectors, $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, PCA is performed, and the resulting eigenvectors are the vectorized eigenmatrices. The task of speaker adaptation is reduced to finding an MLLR transformation for the new speaker, which is assumed to lie in the span of the M leading eigenmatrices (i.e. the M eigenvectors with the largest eigenvalues). Thus, for a new speaker, if his MLLR transformation is $\mathbf{Y}$, then we have

$$vec(\mathbf{Y}) = \mathbf{y} = \sum_{m=1}^{M} w_m \mathbf{v}_m \ , \tag{1}$$

where $\mathbf{w} = [w_1, \ldots, w_M]'$ is the eigenmatrix weight vector, and $\mathbf{v}_m$ is the mth vectorized eigenmatrix. Let $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_d]$ where $\mathbf{y}_r \in \mathbb{R}^{(d+1)}$ is the rth row of $\mathbf{Y}'$ (for $r = 1, \ldots, d$). Then $\mathbf{y}_r$ is given by

$$\mathbf{y}_r = \sum_{m=1}^{M} w_m \mathbf{v}_{mr} \ , \tag{2}$$

where $\mathbf{v}_{mr}$ represents the rth row of the mth eigenmatrix.

[1] In the kernel methods terminology, the original space where raw data reside is called the *input space* and the space to which raw data are mapped is called the *feature space*. In order not to confuse this with the acoustic feature space in speech, the feature space in kernel methods will be simply called the "feature space" but may be sometimes called the "*kernel-induced feature space*" for additional clarity.

Hence, the gth Gaussian mean of the new speaker model is

$$\begin{aligned} \boldsymbol{\mu}_g &= \mathbf{Y}'\boldsymbol{\xi}_g^{(si)} \\ \Rightarrow \mu_{gr} &= \mathbf{y}_r'\boldsymbol{\xi}_g^{(si)} = \sum_{m=1}^{M} w_m(\mathbf{v}_{mr}'\boldsymbol{\xi}_g^{(si)}) \ , \end{aligned} \qquad (3)$$

where μ_{gr} is the rth component of $\boldsymbol{\mu}_g$.

Given the adaptation data $\boldsymbol{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, the eigenmatrix weights can be estimated by maximizing the likelihood of $\boldsymbol{O}$ as in EV adaptation [1, 2]. Mathematically, one finds the optimal $\hat{\mathbf{w}}$ by *maximizing* the following $Q(\mathbf{w})$ function:

$$Q(\mathbf{w}) = -\sum_{g=1}^{G}\sum_{t=1}^{T}\gamma_t(g)(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w}))'\mathbf{C}_g^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w})) \qquad (4)$$

where $\gamma_t(g)$ is the posterior probability of the observation sequence being at the gth Gaussian at time t, and $\mathbf{C}_g$ is the covariance matrix of the gth Gaussian. Differentiating $Q(\mathbf{w})$ w.r.t. each weight, $w_m, m = 1, \ldots, M$, we get

$$\frac{\partial Q(\mathbf{w})}{\partial w_m} = 2\sum_{g=1}^{G}\sum_{t=1}^{T}\gamma_t(g)(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w}))'\mathbf{C}_g^{-1}\frac{\partial \boldsymbol{\mu}_g(\mathbf{w})}{\partial w_m} \ . \qquad (5)$$

By setting the M derivatives to zero, the optimal weights are obtained by solving the system of M linear equations.

3. Kernel EMLLR (KEMLLR) Adaptation

In KEMLLR adaptation, we try to improve EMLLR by exploiting the possible nonlinearity in the speaker transformation (super)vector space. This is achieved by replacing linear PCA by kernel PCA and the use of composite kernel.

3.1. Kernel Eigenmatrices in the Feature Space

Let $k(\cdot, \cdot)$ be the kernel with an associated mapping φ which maps a speaker's transformation vector $\mathbf{y}$ in the input speaker transformation vector space $\mathcal{Y}$ to $\varphi(\mathbf{y})$ in the kernel-induced high dimensional feature space $\mathcal{F}$. Given the set of N speaker transformation vectors $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \in \mathcal{Y}$, their φ-mapped feature vectors are $\{\varphi(\mathbf{y}_1), \ldots, \varphi(\mathbf{y}_N)\} \in \mathcal{F}$. Let the "centered" map be $\tilde{\varphi}$ so that $\tilde{\varphi}(\mathbf{y}) = \varphi(\mathbf{y}) - \bar{\varphi}$ where $\bar{\varphi} = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{y}_i)$. In addition, let $\mathbf{K}$ be the kernel matrix with $\mathbf{K}_{ij} \equiv k(\mathbf{y}_i, \mathbf{y}_j) = \varphi(\mathbf{y}_i)'\varphi(\mathbf{y}_j)$, and $\tilde{\mathbf{K}}$ be its centered version with $\tilde{\mathbf{K}}_{ij} = \tilde{\varphi}(\mathbf{y}_i)'\tilde{\varphi}(\mathbf{y}_j)$.

Kernel PCA may be performed by eigendecomposition on $\tilde{\mathbf{K}}$ as $\tilde{\mathbf{K}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}'$, where $\mathbf{U} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ with $\boldsymbol{\alpha}_i = [\alpha_{i1}, \ldots, \alpha_{iN}]'$, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$. The mth orthonormal eigenvector of the covariance matrix in the feature space $\mathcal{F}$ is given by [10] as

$$\mathbf{v}_m = \sum_{i=1}^{N}\frac{\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{y}_i) \ . \qquad (6)$$

which are our (vectorized) kernel eigenmatrices in the feature space.

Using the leading M eigenmatrices, the centered transformation vector of the new speaker[2] in the feature space $\tilde{\varphi}^{(kemllr)}(\mathbf{y})$ is represented by

$$\tilde{\varphi}^{(kemllr)}(\mathbf{y}) = \sum_{m=1}^{M} w_m\mathbf{v}_m = \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{y}_i) \ . \qquad (7)$$

3.2. Composite Kernel

Eqn. (3) shows that in order to compute the mean vectors of a new speaker, one will need to access each row of his transformation matrix. However, the row information, in general, is lost during the φ-mapping of the transformation vectors to the kernel-induced feature space. To preserve the row information, a composite kernel is used: a possibly different mapping, $\varphi_r, r = 1, \ldots, d$, is used for each row vector of the transformations, and then a composite function is applied. For example, the following direct sum composite kernel has been used in kernel eigenvoice adaptation [7] with good results:

$$k(\mathbf{y}_i, \mathbf{y}_j) = \sum_{r=1}^{d}\varphi_r(\mathbf{y}_{ir})'\varphi_r(\mathbf{y}_{jr}) = \sum_{r=1}^{d}k_r(\mathbf{y}_{ir}, \mathbf{y}_{jr}) \ , \qquad (8)$$

where $\mathbf{y}_{ir}$ represents the rth constituent (i.e. matrix row in our context) of the vector $\mathbf{y}_i$.

Thus, the φ_r-mapping of the rth row of the new speaker's transformation $\mathbf{Y}$ is given by

$$\tilde{\varphi}_r^{(kemllr)}(\mathbf{y}_r) = \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{y}_{ir}) \ . \qquad (9)$$

3.3. Kernel Evaluation

Using Eqn. (9), the similarity between $\varphi_r^{(kemllr)}(\mathbf{y}_r)$ and $\varphi_r(\boldsymbol{\xi}_g^{(si)})$ can be computed as follows:

$$\begin{aligned} & k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)}) \\ \equiv\ & \varphi_r^{(kemllr)}(\mathbf{y}_r))'\varphi_r(\boldsymbol{\xi}_g^{(si)}) && (10) \\ =\ & \left[\left(\sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}_r(\mathbf{y}_{ir})\right) + \bar{\varphi}_r\right]'\varphi_r(\boldsymbol{\xi}_g^{(si)}) \\ =\ & \left[\left(\sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}(\varphi_r(\mathbf{y}_{ir}) - \bar{\varphi}_r)\right) + \bar{\varphi}_r\right]'\varphi_r(\boldsymbol{\xi}_g^{(si)}) \\ =\ & A_r(g) + \sum_{m=1}^{M}\frac{w_m}{\sqrt{\lambda_m}}B_r(m, g) \ , && (11) \end{aligned}$$

where $\bar{\varphi}_r = \frac{1}{N}\sum_{i=1}^{N}\varphi_r(\mathbf{y}_{ir})$ is the rth part of $\bar{\varphi}$,

$$A_r(g) \quad = \quad \bar{\varphi}_r'\varphi_r(\boldsymbol{\xi}_g^{(si)}) = \frac{1}{N}\sum_{i=1}^{N}k_r(\mathbf{y}_{ir}, \boldsymbol{\xi}_g^{(si)}), \qquad (12)$$

and

$$B_r(m, g) = \sum_{i=1}^{N}\alpha_{mi}(k_r(\mathbf{y}_{ir}, \boldsymbol{\xi}_g^{(si)}) - A_r(g)). \qquad (13)$$

Furthermore, the derivative of $k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})$ w.r.t. each eigenvoice weight $w_m,\ m = 1, \ldots, M$, is given by

$$\frac{\partial}{\partial w_m}\left(k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})\right) = \frac{B_r(m, g)}{\sqrt{\lambda_m}} \ , \qquad (14)$$

[2] The notation of the transformation vector in the feature space requires some explanation. In kernel methods, the existence of an object in the feature space does not necessarily imply the existence of its pre-image in the input space. Here, we use $\varphi^{(kemllr)}(\mathbf{y})$ to represent the image even if the pre-image $\mathbf{y}$ may not exist due to the intuitiveness of the notation. Notice that our KEMLLR adaptation does not require the existence of the pre-image $\mathbf{y}$ in the input transformation (super)vector space.

which will be needed for the maximum likelihood estimation of the eigenmatrix weights.

Notice that all the kernel values in Eqns. (12,13) may be computed offline prior to adaptation.

3.4. Gradient of Gaussian Means

Eqn. (5) requires the gradient of $\mu_g^{(kemllr)}$ w.r.t. each eigenmatrix weight $w_m, m = 1, \ldots, M$. Since the eigenmatrices reside in the kernel-induced feature space and *not* in the acoustic observation space, we have to relate $\mu_{gr}^{(kemllr)}$ with $\mathbf{w}$ via the kernel values $k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})$. This can be done if we have a kernel function that is a function of the inner product of its inputs since $\mu_{gr} = \mathbf{y}_r' \boldsymbol{\xi}_g^{(si)}$. That is, we need a kernel function k_r such that $k_r(\mathbf{u}, \mathbf{v}) = F(\mathbf{u}'\mathbf{v})$, where F is invertible. Then,

$$\mu_{gr}^{(kemllr)} = \mathbf{y}_r' \boldsymbol{\xi}_g^{(si)} = F^{-1}(k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})) \; , \qquad (15)$$

which, in turn, is a function of $\mathbf{w}$ as given by Eqn. (11), and its derivative w.r.t. w_m can be readily obtained.

3.4.1. *Gaussian Kernels*

Let's consider the following Gaussian kernel

$$k_r(\mathbf{u}, \mathbf{v}) = \exp(-\beta_r \|\mathbf{u} - \mathbf{v}\|^2) \; . \qquad (16)$$

The Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$ is given by

$$\|\mathbf{u} - \mathbf{v}\|^2 = -\frac{1}{\beta_r} \log k_r(\mathbf{u}, \mathbf{v}) \; . \qquad (17)$$

Since

$$\begin{aligned} \|\mathbf{u} - \mathbf{v}\|^2 &= \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\mathbf{u}'\mathbf{v} \\ \Rightarrow \quad \mathbf{u}'\mathbf{v} &= \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - \|\mathbf{u} - \mathbf{v}\|^2) \; , \end{aligned} \qquad (18)$$

and in our case, $\mathbf{u} = \mathbf{y}_r$ and $\mathbf{v} = \boldsymbol{\xi}_g^{(si)}$, therefore, we have

$$\mu_{gr}^{(kemllr)} = \frac{1}{2} \left[\|\boldsymbol{\xi}_g^{(si)}\|^2 + \frac{1}{\beta_r} \log \left(\frac{k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})}{k_r^{(kemllr)}(\mathbf{y}_r, \mathbf{0})} \right) \right] \; . \qquad (19)$$

Substituting Eqns. (11,12,13) into Eqn. (19), differentiating the result w.r.t. w_m, and making use of the gradient in Eqn.(14), we get $\dfrac{\partial \mu_{gr}^{(kemllr)}}{\partial w_m}$

$$= \frac{1}{2\beta_r \sqrt{\lambda_m}} \left[\frac{B_r(m, g)}{k_r^{(kemllr)}(\mathbf{y}_r, \boldsymbol{\xi}_g^{(si)})} - \frac{B_r(m, -1)}{k_r^{(kemllr)}(\mathbf{y}_r, \mathbf{0})} \right] \; , \qquad (20)$$

where we use the index $g = -1$ to represent a special augmented vector $\boldsymbol{\xi}_{-1}^{(si)}$ which is the zero vector $\mathbf{0}$.

3.5. ML Estimation of Eigenmatrix Weights

Using Eqn. (20), the derivatives of $Q(\mathbf{w})$ of Eqn. (5) w.r.t each of the M weights $w_m, m = 1, \ldots, M$, can be obtained. However, due to the nonlinearity of the kernel functions, there is no closed form solution for the optimal $\mathbf{w}$. Instead, the eigenmatrix weights are found using Gradient Ascent algorithm. That is, the estimate of $\mathbf{w}$ at the nth Gradient Ascent iteration is updated using the learning rate $\eta(n)$ as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n) \frac{\partial Q(\mathbf{w})}{\partial \mathbf{w}} \; .$$

3.6. Robust KEMLLR

When the amount of adaptation data is really small, the MLLR transformation found by KEMLLR may not be reliable. To get a more robust estimate , the transformation found by KEMLLR is interpolated with the identity matrix. Equivalently, a mean vector found by KEMLLR is interpolated with the corresponding SI mean vector as follows:

$$\mu_{gr}^{(rkemllr)} = w_0 \mu_{gr}^{(si)} + (1 - w_0) \mu_{gr}^{(kemllr)} \; , \; 0 \le w_0 \le 1.0 \; . \quad (21)$$

And the gradients of the Gaussian means are updated as below:

$$\frac{\partial \mu_{gr}^{(rkemllr)}}{\partial w_0} = \mu_{gr}^{(si)} - \mu_{gr}^{(kemllr)} \; , \qquad (22)$$

and

$$\frac{\partial \mu_{gr}^{(rkemllr)}}{\partial w_m} = (1 - w_0) \frac{\partial \mu_{gr}^{(kemllr)}}{\partial w_m} \; , \quad m = 1, \ldots, M. \quad (23)$$

4. Experimental Evaluation

The proposed KEMLLR speaker adaptation method was evaluated on the DARPA Resource Management continuous speech database RM1. RM1 comprises a speaker-independent (SI) section and a speaker-dependent (SD) section. The SI section consists of 3990 training utterances from 109 speakers. On the other hand, there are 12 speakers in the SD section, each having 600 utterances for training, 100 utterances for development, and 100 utterances for evaluation.

4.1. Feature Extraction and Acoustic Modeling

As a preliminary investigation of our new KEMLLR adaptation method, the following simple acoustic vectors and acoustic models were used. Forty-seven context-independent phoneme models were trained using the SI training set. Each phoneme model was a strictly left-to-right 3-state hidden Markov model (HMM) with 10 Gaussian mixtures per state. In addition, there were a 1-state short pause model and a 3-state silence model. The acoustic vector has a dimension $d = 13$, consisting of 12 MFCCs and the normalized log energy extracted from speech frames of 25 ms long at the frame rate of 100Hz.

4.2. Experimental Procedure

From the SI model, an SD model was constructed for each of the 109 speakers in the SI training set using global MLLR adaptation. As a result, we obtained a set of $N = 109$ transformation vectors for deriving the kernel eigenmatrices. For each of the 12 speakers in the SD section, 3 sets of adaptation data were randomly chosen from his 100 development utterances so that each adaptation set was about 4–5s long, consisting of 2–3 utterances. Each adaptation method was run on each of the 3 adaptation sets of each speaker, and the resulting adapted models were tested on his 100 evaluation utterances using word-pair grammar. Reported results are the average of all adaptation sets of all speakers.

The following models or adaptation methods are compared:

SI: speaker-independent model.

MLLR: MLLR adaptation.

EMLLR: eigenspace-based MLLR adaptation.

KEMLLR: kernel EMLLR adaptation.

Table 1: Adaptation performance of the SI model, MLLR, EMLLR, and KEMLLR adaptation on the evaluation set of the SD section of RM1.

Model/Adaptation	Word Accuracy	WER Reduction
SI	78.27%	—
MLLR	78.31%	0.18%
EMLLR	78.72%	2.07%
KEMLLR	80.63%	10.86%

MLLR adaptation was done using the HTK software with a global diagonal transformation. EMLLR was implemented using KEMLLR with linear kernel, and all EMLLR and KEMLLR models were interpolated with the SI model as said in Section 3.6. Some of the experimental parameters for KEMLLR were initialized or set empirically as follows (which are by no means optimal):

- w_0 was initialized to 0.5.
- The eigenmatrix weights $w_m, m = 1, \ldots, M$, were initialized by projecting the following transformation,

$$\mathbf{Y}^{(si)} = \begin{bmatrix} 1 & \mathbf{0} & \cdots & \mathbf{0} & 0 \\ \mathbf{0} & 1 & \cdots & \mathbf{0} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 0 \end{bmatrix} . \tag{24}$$

  onto each of the M kernel eigenmatrices.
- $\beta_r = \beta = 0.0001$ for $r = 1, \ldots, d$
- The learning rate for Gradient Ascent was initialized to 1.0×10^{-9} which was adjusted dynamically depending on the increase or decrease in the Q function value after each iteration.
- Pilot experiments on the development data found that $M = 50$ eigenmatrices gave good adaptation results.

4.3. Evaluation Results

The comparative performance of the various adaptation methods and the SI model is shown in Table 1. It can be seen that for this particular task, the performance of the various adaptation methods and SI model is ranked in the following increasing order: SI $\approx$ MLLR $<$ EMLLR $<$ KEMLLR. The improvement of KEMLLR over EMLLR shows that the use of kernel PCA with composite kernel is effective in deriving better eigenmatrices for adaptation.

5. Conclusions and Future Work

In this paper, we propose an improvement to the eigenspace-based MLLR (EMLLR) adaptation method by deriving the eigenmatrices using kernel methods. In our novel *kernel EMLLR (KEMLLR)* adaptation, kernel principal component analysis is used to exploit possible nonlinearity in the transformation (super)vector space, and composite kernel is used to preserve the row information in a transformation. Preliminary adaptation experiments on RM1 shows that KEMLLR outperformed MLLR and EMLLR, and reduced the word error rate of the speaker-independent model by 11%.

KEMLLR adaptation is a nonlinear optimization problem; it is currently solved by Gradient Ascent method and is relatively slower. As a result, for this preliminary investigation, we used simple acoustic vectors and models with a global transformation to create each speaker-dependent (SD) model. We are finding ways to speed up the adaptation process, and will evaluate the method again using cross-word triphone models and multiple transformations in the estimation of the SD models.

6. Acknowledgements

This research is partially supported by the Research Grants Council of the Hong Kong SAR under the grant numbers HKUST6201/02E and CA02/03.EG04.

7. References

[1] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 695–707, Nov 2000.

[2] K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee, “Fast speaker adaptation using eigenspace-based maximum likelihood linear regression,” in *Proceedings of the International Conference on Spoken Language Processing*, 2000, vol. 3, pp. 742–745.

[3] R. Kuhn, F. Perronnin, P. Nguyen, J. C. Junqua, and L. Rigazio, “Very fast adaptation with a compact context-dependent eigenvoice model,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[4] B. Zhou and J. Hansen, “A novel algorithm for rapid speaker adaptation based on structural maximum likelihood eigenspace mapping,” in *Proceedings of the European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sept. 2001, vol. 2, pp. 1215–1218.

[5] J. L. Gauvain and C. H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, April 1994.

[6] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Journal of Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[7] J. T. Kwok, B. Mak, and S. Ho, “Eigenvoice speaker adaptation via composite kernel PCA,” in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.

[8] B. Mak, J. T. James, and S. Ho, “A study of various composite kernels for kernel eigenvoice speaker adaptation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004.

[9] B. Schölkopf and A.J. Smola, *Learning with Kernels*, MIT, 2002.

[10] B. Schölkopf, A. Smola, and K. R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.