KERNEL EIGENSPACE-BASED MLLR ADAPTATION USING MULTIPLE REGRESSION CLASSES

Roger Hsiao and *Brian Mak*

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

ABSTRACT

Recently, we have been investigating the application of kernel methods to improve the performance of eigenvoice-based adaptation methods by exploiting possible nonlinearity in their original working space. We proposed the *kernel eigenvoice adaptation* (KEV) in [1], and the *kernel eigenspace-based MLLR adaptation* (KEMLLR) in [2]. In KEMLLR, speaker-dependent MLLR transformation matrices are mapped to a kernel-induced high dimensional feature space, and kernel principal component analysis (KPCA) is used to derive a set of eigenmatrices in the feature space. A new speaker is then represented by a linear combination of the leading eigenmatrices. In this paper, we further improve KEMLLR by the use of multiple regression classes and the quasi-Newton BFGS optimization algorithm.

1. INTRODUCTION

When the amount of adaptation speech is really small, say, a few seconds, eigenvoice-based adaptation methods [3, 4, 5] have been shown more effective than the traditionally more popular methods such as the Bayesian-based MAP adaptation [6] and the transformation-based MLLR adaptation [7]. Eigenspace-based MLLR (EMLLR) adaptation [4] is a variant of the standard EV adaptation [3]. Instead of finding a small set of eigenvoices (EV) 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 EMLLR adaptation called *kernel eigenspace-based MLLR adaptation* (KEMLLR) [2] by exploiting possible nonlinearity in the MLLR transformation supervector space using kernel methods [8]. The basic idea is to map the speakers' MLLR transformation supervectors to a high dimensional feature space via some nonlinear map, and then apply principal component analysis (PCA) 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 (KPCA). The computational procedure depends only on the inner products in the feature space, which can be obtained efficiently with a suitable kernel function. 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 [1].

In this paper, we further improve KEMLLR by the use of multiple regression classes and the more advanced quasi-Newton BFGS optimization algorithm.

2. EIGENSPACE-BASED MLLR (EMLLR) ADAPTATION

Suppose there is a set of N speaker-dependent (SD) hidden Markov models (HMMs) of the same topology with mixture Gaussian states. These SD models are estimated from the SI model by MLLR transformation using L regression classes. Let H be the mapping function that maps the gth Gaussian to its regression class $h = H(g)$ where $h \in \{1, \ldots, L\}$. Thus, the gth Gaussian mean vector $\boldsymbol{\mu}_g^{(i)} \in \mathbb{R}^d$ of the ith speaker is given by

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

where $\mathbf{Y}_{H(g)}^{(i)} \in \mathbb{R}^{d \times (d+1)}$ is his MLLR transformation for the $H(g)$-th regression class, 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 supervector* (STSV) is obtained by stacking up the L vectorized MLLR transformation matrices, $\mathbf{Y}_1^{(i)}, \ldots, \mathbf{Y}_L^{(i)}$. Let's denote the STSV of the ith speaker by $\mathbf{y}^{(i)} = [vec(\mathbf{Y}_1^{(i)})', \ldots, vec(\mathbf{Y}_L^{(i)})']'$. From the N STSVs, $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(N)}\}$, PCA is performed to obtain the eigenvectors which are the vectorized eigenmatrices. For a new speaker, his STSV is approximated as a linear combination of the leading M vectorized eigenmatrices as $\mathbf{y} = \sum_{m=1}^{M} w_m \mathbf{v}_m$, where $\mathbf{w} = [w_1, \ldots, w_M]'$ is the eigenmatrix weight vector, and $\mathbf{v}_m$ is the mth vectorized eigenmatrix.

Let $\mathbf{y} = [\ldots, \mathbf{y}'_{h1}, \ldots, \mathbf{y}'_{hd}, \ldots]'$ where $\mathbf{y}_{hr} \in \mathbb{R}^{(d+1)}$ is the rth row of the hth MLLR transformation matrix (for $r = 1, \ldots, d$ and $h = 1, \ldots, L$). Then $\mathbf{y}_{hr}$ is given by $\mathbf{y}_{hr} = \sum_{m=1}^{M} w_m \mathbf{v}_{mhr}$, where $\mathbf{v}_{mhr}$ represents the rth row of the hth transformation matrix embedded in the mth eigenvector.

Hence, the gth Gaussian mean of the new speaker model, which belongs to the hth regression class (as $h = H(g)$), is

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

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

0-7803-8874-7/05/$20.00 ©2005 IEEE

$\boldsymbol{O}$ [3, 4], or, equivalently the following $Q(\mathbf{w})$ function:

$$Q(\mathbf{w}) = -\sum_{g=1}^{G}\sum_{t=1}^{T}\gamma_t(g)(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w}))'\mathbf{C}_g^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w})) \quad (2)$$

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

$$\frac{\partial Q(\mathbf{w})}{\partial w_m} = 2\sum_{g=1}^{G}\sum_{t=1}^{T}\gamma_t(g)(\mathbf{o}_t - \boldsymbol{\mu}_g(\mathbf{w}))'\mathbf{C}_g^{-1}\frac{\partial \boldsymbol{\mu}_g(\mathbf{w})}{\partial w_m} . \quad (3)$$

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 supervector 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 STSV space to $\varphi(\mathbf{y})$ in the kernel-induced high dimensional feature space. Given the set of N STSVs $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, their φ-mapped feature vectors are $\{\varphi(\mathbf{y}_1), \ldots, \varphi(\mathbf{y}_N)\}$. Let $\tilde{\mathbf{K}}$ be the centered kernel matrix with $\tilde{\mathbf{K}}_{ij} \equiv \tilde{k}(\mathbf{y}_i, \mathbf{y}_j) = \tilde{\varphi}(\mathbf{y}_i)'\tilde{\varphi}(\mathbf{y}_j)$ where $\tilde{\varphi}(\mathbf{y}) = \varphi(\mathbf{y}) - \bar{\varphi}$ and $\bar{\varphi} = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{y}_i)$.

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 $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$. Using the leading M eigenmatrices of the covariance matrix in the kernel-induced feature space, the centered STSV of the new speaker in the feature space $\tilde{\varphi}^{(kemllr)}(\mathbf{y})$ is given by

$$\tilde{\varphi}^{(kemllr)}(\mathbf{y}) = \sum_{m=1}^{M} w_m \mathbf{v}_m = \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}(\mathbf{y}_i) . \quad (4)$$

3.2. Composite Kernel

Analogous to the use of composite kernels to preserve the state information in kernel eigenvoice [1], the row information of each transformation matrix is preserved in KEMLLR using the direct sum composite kernel so that

$$k(\mathbf{y}_i, \mathbf{y}_j) = \sum_{h=1}^{L}\sum_{r=1}^{d} k_{hr}(\mathbf{y}_{ihr}, \mathbf{y}_{jhr}) , \quad (5)$$

where $\mathbf{y}_{ihr}$ represents the part of $\mathbf{y}_i$ corresponding to the rth row of the MLLR transformation matrix of the hth regression class before the φ-mapping.

Thus, the φ_{hr}-mapping of the rth row of the MLLR transform of the hth regression class for the new speaker's STSV is given by

$$\tilde{\varphi}_{hr}^{(kemllr)}(\mathbf{y}_{hr}) = \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{w_m\alpha_{mi}}{\sqrt{\lambda_m}}\tilde{\varphi}_{hr}(\mathbf{y}_{ihr}) . \quad (6)$$

3.3. Kernel Evaluation

Using Eqn. (6), the similarity between $\varphi_{hr}^{(kemllr)}(\mathbf{y}_{hr})$ and $\varphi_{hr}(\boldsymbol{\xi}_g^{(si)})$ can be computed as follows:

$$\begin{aligned} k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \boldsymbol{\xi}_g^{(si)}) &\equiv \varphi_{hr}^{(kemllr)}(\mathbf{y}_{hr}))'\varphi_{hr}(\boldsymbol{\xi}_g^{(si)}) \\ &= A_{hr}(g) + \sum_{m=1}^{M}\frac{w_m}{\sqrt{\lambda_m}}B_{hr}(m, g) , \quad (7) \end{aligned}$$

where

$$A_{hr}(g) = \bar{\varphi}_{hr}'\varphi_{hr}(\boldsymbol{\xi}_g^{(si)}) = \frac{1}{N}\sum_{i=1}^{N}k_{hr}(\mathbf{y}_{ihr}, \boldsymbol{\xi}_g^{(si)}), \quad (8)$$

$$B_{hr}(m, g) = \sum_{i=1}^{N}\alpha_{mi}(k_{hr}(\mathbf{y}_{ihr}, \boldsymbol{\xi}_g^{(si)}) - A_{hr}(g)) , \quad (9)$$

and $\bar{\varphi}_{hr} = \frac{1}{N}\sum_{i=1}^{N}\varphi_{hr}(\mathbf{y}_{ihr})$. Notice that all the kernel values in Eqns. (8,9) may be computed offline prior to adaptation.

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

$$\frac{\partial}{\partial w_m}\left(k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \boldsymbol{\xi}_g^{(si)})\right) = \frac{B_{hr}(m, g)}{\sqrt{\lambda_m}} . \quad (10)$$

3.4. Gradient of Gaussian Means

Eqn. (3) requires the gradient of $\mu_g^{(kemllr)}$ w.r.t. each eigenmatrix weight $w_m, m = 1, \ldots, M$. This can be obtained by using Gaussian kernels for the composite kernels,

$$k_{hr}(\mathbf{u}, \mathbf{v}) = \exp(-\beta_{hr}\|\mathbf{u} - \mathbf{v}\|^2) ,$$

and the identity $\mathbf{u}'\mathbf{v} = \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - \|\mathbf{u} - \mathbf{v}\|^2)$. By letting $\mathbf{u} = \mathbf{y}_{hr}$ and $\mathbf{v} = \boldsymbol{\xi}_g^{(si)}$, we have

$$\mu_{gr}^{(kemllr)} = \frac{1}{2}\left[\|\boldsymbol{\xi}_g^{(si)}\|^2 + \frac{1}{\beta_{hr}}\log\left(\frac{k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \boldsymbol{\xi}_g^{(si)})}{k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \mathbf{0})}\right)\right] . \quad (11)$$

Substituting Eqns. (7,8,9) into Eqn. (11), differentiating the result w.r.t. w_m, and making use of the gradient in Eqn.(10), we get $\dfrac{\partial \mu_{gr}^{(kemllr)}}{\partial w_m}$

$$= \frac{1}{2\beta_{hr}\sqrt{\lambda_m}}\left[\frac{B_{hr}(m, g)}{k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \boldsymbol{\xi}_g^{(si)})} - \frac{B_{hr}(m, -1)}{k_{hr}^{(kemllr)}(\mathbf{y}_{hr}, \mathbf{0})}\right] , \quad (12)$$

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 by the Quasi-Newton BFGS Method

Using Eqn. (12), the derivatives of $Q(\mathbf{w})$ of Eqn. (3) 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 $\mathbf{w}$. In the past [2], the weights are obtained by gradient ascent method and we notice that sometimes it is not effective and gets stuck. Now we replace it by the quasi-Newton BFGS optimization algorithm which consistently gives better solutions. Quasi-Newton method is similar to the traditional Newton's method and makes use of the Hessian to retrieve the

Newton's direction. However, it approximates the Hessian with an estimate that can be derived solely from the gradient. As a result, it is more efficient and it can enforce the Hessian estimate to be strictly positive-definite.

In quasi-Newton method, the inverse of the Hessian matrix $\mathbf{A}^{-1}$ is approximated by $\mathbf{H}_i$ in an iterative procedure so that $\lim_{i \to \infty} \mathbf{H}_i = \mathbf{A}^{-1}$, where $\mathbf{H}_i$ is the Hessian inverse in the ith iteration, and it has to be positive definite and symmetric. In this paper, $\mathbf{H}_i$ is updated by the (BFGS) algorithm as follows:

$$\mathbf{H}_{i+1} = (I - \frac{\mathbf{s}_i \mathbf{y}_i^{'}}{\mathbf{y}_i^{'} \mathbf{s}_i})\mathbf{H}_i(I - \frac{\mathbf{y}_i \mathbf{s}_i^{'}}{y_i^{'} \mathbf{s}_i}) + \frac{\mathbf{s}_i \mathbf{s}_i^{'}}{\mathbf{y}_i^{'} \mathbf{s}_i} \quad (13)$$

where,

$$\mathbf{s}_i = \mathbf{w}_{i+1} - \mathbf{w}_i \quad (14)$$

$$\mathbf{y}_i = \bigtriangledown Q(\mathbf{w}_{i+1}) - \bigtriangledown Q f(\mathbf{w}_i) \quad (15)$$

Detailed description and proof are available in [9].

Finally, the optimal eigenmatrix weights can be optimized iteratively by the following updating formula:

$$\mathbf{w}_{i+1} = -\lambda \mathbf{H}_i \bigtriangledown Q(\mathbf{w})|_{\mathbf{w}_i}$$

where λ is a learning rate to be determined by a line search algorithm, and the gradient can be computed from Eqns. (3,12).

3.6. Robust KEMLLR

To get a more robust estimate when the amount of adaptation data is really small, we proposed in [2] to interpolate the transformations found by KEMLLR with the identity matrix. Equivalently, a mean vector found by KEMLLR is interpolated with the corresponding SI mean vector. The interpolation weight w_0 is optimized with all other eigenmatrix weights.

4. EXPERIMENTAL EVALUATION

The proposed KEMLLR speaker adaptation method was evaluated on the DARPA Resource Management continuous speech database RM1. RM1 consists of 3990 SI training utterances from 109 speakers, and 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

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 MLLR adaptation and the number of regression classes were varied. As a result, we obtained a set of $N = 109$ transformation supervectors for deriving the kernel eigenmatrices. Experiments were performed with either 5s or 10s adaptation data. To improve reliability of the results, for each test speaker, 3 sets of adaptation data were randomly chosen from his 100 development utterances. All reported results are the averages of experiments over the 3 adaptation sets of all speakers, and the adapted models were tested on their 100 evaluation utterances using word-pair grammar.

The following models or adaptation methods are compared:

SI: speaker-independent model.

MLLR-D: MLLR adaptation with diagonal transformation.

MLLR-F: MLLR adaptation with full transformation.

EMLLR: eigenspace-based MLLR adaptation.

KEMLLR: kernel EMLLR adaptation.

MLLR adaptation was done using the HTK software with diagonal or full transformation with (a maximum of) 32 regression classes. However, by default, HTK requires at least 700 frames of speech for each regression class. As some configurations had very few data, this threshold was lowered in order to force HTK to perform MLLR. 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.

4.3. Number of Eigenmatrices and Regression Classes

Fig. 1 and Fig. 2 describe the complicated relationship among the number of eigenmatrices, number of regression classes, and the amount of data used in EMLLR or KEMLLR adaptation. Twenty-five, 50, 75, and 109 eigenmatrices were tried with one or two regression classes using 5s or 10s of adaptation speech. As expected, better adaptation performance results when more adaptation data are available. Notice that when EMLLR or KEMLLR are done with multiple regression classes, they can perform PCA or KPCA separately on each class or on the concatenated transformation supervectors. In our preliminary investigation, the former always gave better adaptation performance than the latter. As a consequence, all experiments reported here when multiple regression classes were used treated them separately. We have the following observations:

- On the one hand, more regression classes should give more detailed modeling and should give better results. On the other hand, more regression classes require more adaptation data as there are more weights to estimate. The effect is more pronounced for KEMLLR: with 2 regression classes, the performance actually drops with only 5s of adaptation speech, but is elevated when 10s of adaptation speech is provided.
- KEMLLR generally outperforms EMLLR adaptation when the same number of eigenmatrices and regression classes are employed using the same amount of adaptation speech. This shows that the leading eigenmatrices derived in KEMLLR using KPCA are more effective in capturing useful speaker information.
- When all eigenmatrices are employed, EMLLR adaptation performance may still improve. This may suggest that there are residual nonlinear information which cannot be covered by the leading eigenmatrices derived by PCA in EMLLR so that using all eigenmatrices may still improve the performance. However, this is not true for KEMLLR where using all kernel eigenmatrices will degrade the performance.

That suggests that the trailing kernel eigenmatrices are really noises. Thus, once again, KPCA helps to extract the nonlinear eigen-information more effective than PCA.

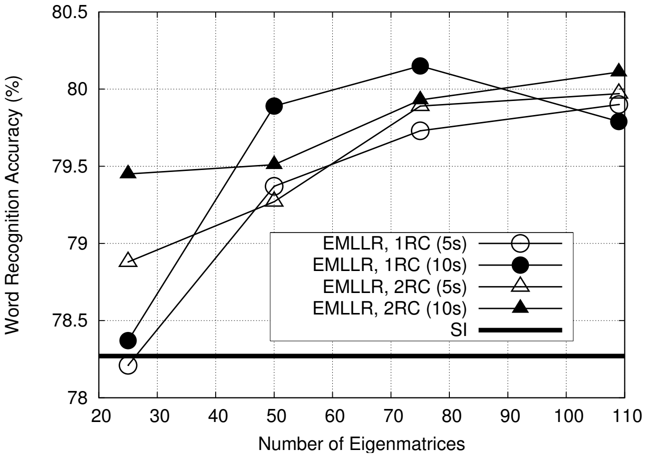

Fig. 1. Performance of EMLLR adaptation on RM with a 10-mixture HMM.

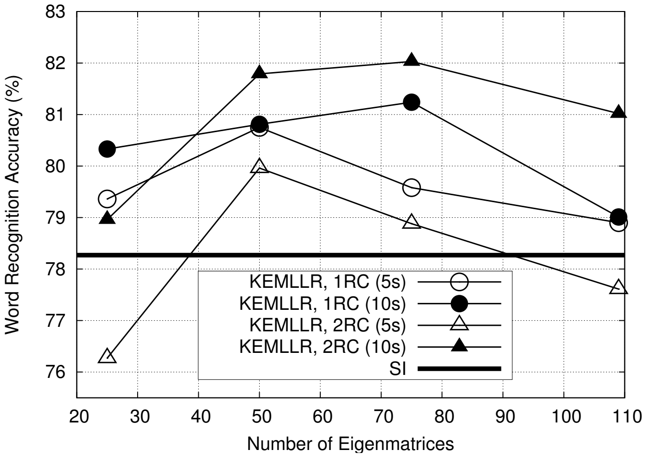

Fig. 2. Performance of KEMLLR adaptation on RM with a 10-mixture HMM.

4.4. Comparison among Various Adaptation Methods.

In this experiment, the SI model, MLLR-D, MLLR-F, EMLLR, and KEMLLR are compared at their best settings. The results are summarized in Table 1. It is found that when only 5s of adaptation speech were available, even we lowered the threshold for MLLR-D and MLLR-F when we ran HTK, they still could not be run. On the other hand, EMLLR successfully reduced the word error rate (WER) by 7.82%, and KEMLLR could reduce the WER by 11.4%. When 10s of adaptation speech were provided, MLLR-F became effective and matched the performance of KEMLLR. EMLLR again does not perform as well as KEMLLR, and MLLR-D gave the least performance gain.

Table 1. Performance of the SI model, MLLR, EMLLR, eKEV and KEMLLR adaptation on RM with a 10-mixture HMM.

Model/Adaptation	Word Accuracy	
	5s	10s
SI	78.27%	78.27%
MLLR-D	N/A	78.90%
MLLR-F	N/A	82.10%
EMLLR	79.97%	80.11%
KEMLLR	80.75%	82.03%

5. CONCLUSIONS

In this paper, we improve kernel eigenspace-based MLLR (KEMLLR) adaptation method further by using multiple regression classes, and investigated the relationship among the number of kernel eigenmatrices, number of regression classes, and the amount of adaptation data. We show that when only 4–5s of speech are used, both EMLLR and KEMLLR are effective, but KEMLLR gives greater performance improvement than EMLLR. When 10s of speech are available, KEMLLR performance is than matched by standard MLLR using full transformation. All in all, KEMLLR seems to be effective for fast speaker adaptation using less than 10s of adaptation speech.

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] B. Mak, J. T. Kwok, and S. Ho, "A study of various composite kernels for kernel eigenvoice speaker adaptation," in *ICASSP*, Montreal, Canada, 2004, vol. I, pp. 325–328.

[2] B. Mak and R. Hsiao, "Improving eigenspace-based MLLR adaptation by kernel PCA," in *ICSLP*, Jeju Island, South Korea, 2004.

[3] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. on SAP*, vol. 8, no. 4, pp. 695–707, Nov 2000.

[4] K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihood linear regression," in *ICSLP*, 2000, vol. 3, pp. 742–745.

[5] R. Kuhn, F. Perronnin, P. Nguyen, J. C. Junqua, and L. Rigazio, "Very fast adaptation with a compact context-dependent eigenvoice model," in *ICASSP*, May 2001, vol. 1, pp. 373–376.

[6] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. on SAP*, vol. 2, no. 2, pp. 291–298, April 1994.

[7] 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.

[8] B. Schölkopf and A.J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[9] J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer-Verlag, Berlin Heidelberg, 2003.