Maximum Penalized Likelihood Kernel Regression for Fast Adaptation

Brian Kan-Wing Mak, *Member, IEEE*, Tsz-Chung Lai, Ivor W. Tsang, and James Tin-Yau Kwok, *Senior Member, IEEE*

***Abstract*—This paper proposes a nonlinear generalization of the popular *maximum-likelihood linear regression* (MLLR) adaptation algorithm using kernel methods. The proposed method, called *maximum penalized likelihood kernel regression* adaptation (MPLKR), applies kernel regression with appropriate regularization to determine the affine model transform in a kernel-induced high-dimensional feature space. Although this is not the first attempt of applying kernel methods to conventional linear adaptation algorithms, unlike most of other kernelized adaptation methods such as kernel eigenvoice or kernel eigen-MLLR, MPLKR has the advantage that it is a convex optimization and its solution is always guaranteed to be globally optimal. In fact, the adapted Gaussian means can be obtained analytically by simply solving a system of linear equations. From the Bayesian perspective, MPLKR can also be considered as the kernel version of *maximum a posteriori linear regression* (MAPLR) adaptation. Supervised and unsupervised speaker adaptation using MPLKR were evaluated on the Resource Management and Wall Street Journal 5K tasks, respectively, achieving a word error rate reduction of 23.6% and 15.5% respectively over the speaker-independently model.**

***Index Terms*—Kernel regression, maximum-likelihood linear regression (MLLR), reference speaker weighting, speaker adaptation.**

I. Introduction

In general, there is a performance gap between a well-trained speaker-dependent (SD) model and a speaker-independent (SI) model on recognizing speech from a specific speaker. Nevertheless, it is impractical to require a speaker to provide a large amount of speech to train a good SD model for himself/herself. That leaves one to start with an SI model, and try to accommodate the speaker's acoustic characteristics to work with the already trained SI model via adaptation techniques using a relatively small amount of speech data from the new speaker. In feature-based adaptation, such as piecewise linear acoustic mapping [1], and *feature-space maximum-likelihood linear regression*.[1] (fMLLR) [2], [3], the testing acoustic features are modified to match closer to the training acoustics. In model-based adaptation, the SI model parameters are modified so that the adapted model fits better the new speaker. There are three basic categories of speaker adaptation methods: speaker-clustering-based methods [4]–[6] (including the eigenspace-based methods [7], [8]), Bayesian-based methods such as the *maximum a posteriori* (MAP) adaptation [9], and transformation-based methods, most notably, *maximum-likelihood linear regression* (MLLR) adaptation [10].

In our experience, the plain MLLR together with a carefully built regression tree works well in most cases: The algorithm is simple, has an analytical solution which is globally optimal, and naturally improves with more adaptation data through the use of regression tree. However, for many online applications over the telephone, for instance, directory or other query services in which the users say only a few words and no pre-registration is required, the amount of adaptation data can be very limited—typically less than 10 s of speech—the MLLR transforms can be easily overtrained. In such cases, speaker-clustering-based or eigenspace-based adaptation methods such as eigenvoice [7], eigen-MLLR [8], cluster weighting [6], or reference speaker weighting [6], [11] usually gives better adaptation performance. On the other hand, by imposing various constraints on the MLLR transformation, the original MLLR adaptation method can be modified for fast adaptation. Some notable efforts are summarized as follows.

- Reduce the number of free parameters by constraining the MLLR transforms to be block-diagonal or diagonal matrices. Although it results in some improvement, it is an *ad hoc* solution that does not utilize the correlation among all different components of the acoustic feature vector.
- Constrain the solution space of MLLR with a prior distribution in the Bayesian MAP approach, resulting in MAPLR [12]. The first MAPLR algorithm does not really work for very small amount of adaptation data since a robust estimation of the hyperparameters of the prior distribution also requires a fairly substantial amount of adaptation data. Structural MAPLR (SMAPLR) [13] was later proposed to solve the problem by using hierarchical priors as in structural MAP (SMAP) [14]. SMAPLR estimates the prior density in a child node as the posterior density of its parent node. On the other hand, [15] shows that on a task with only 2.5 s of adaptation speech, using the solution from cluster weighting adaptation as the prior

Manuscript received June 24, 2008; revised March 01, 2009. Current version published July 17, 2009. This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grants 617406, 617507, and 614508. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gerhard Rigoll.

B. Mak, T.-C. Lai, and J. T. Kwok are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: mak@cse.ust.hk; kimo@cse.ust.hk; jamesk@cse.ust.hk).

I. W. Tsang is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ivortsang@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2009.2019920

[1]fMLLR transformation, however, is not a feature-space transformation because of the additional Jacobian term per transform as explained in [2], which actually calls the method *constrained MLLR*.

mean for MAPLR can further improve MAPLR performance.

- Regularization technique interpolates the MLLR estimate with a more robust estimate that is obtained from additional data or knowledge sources which is not necessarily the prior distribution. For instance, in *discounted likelihood linear regression* (DLLR) [16], part of the MLLR likelihood is discounted and interpolated with the likelihood computed from the *desired family of distributions* that may explain the adaptation data. It was shown that DLLR gave better performance than diagonal, block-diagonal, or full MLLR in Switchboard with as little as 5 s of adaptation speech.

In this paper, we propose a nonlinear generalization of MLLR for fast adaptation which is referred to as *maximum penalized likelihood kernel regression* (MPLKR) adaptation.[2] MPLKR performs nonlinear regression between the maximum-likelihood (ML) adapted mean vectors and the SI mean vectors with the use of kernel methods [18]–[20]. The basic idea is to first map the SI mean vectors to a high-dimensional feature space via some nonlinear map φ before performing linear regression with appropriate regularization to find the affine model transform. The computational procedure depends only on the inner products of the mapped SI mean vectors in the high-dimensional feature space, which can be obtained efficiently with a suitable kernel function. One attraction of MPLKR is that except for the nonlinear mapping of SI mean vectors, all remaining operations are linear. Thus, the MPLKR transform can be obtained by solving a system of linear equations in much the same way as the original MLLR. It is in contrast with our previous nonlinear extensions to eigenvoice and eigen-MLLR (called kernel eigenvoice (KEV) and kernel eigen-MLLR (KEMLLR) respectively) that are nonlinear optimizations and require gradient-based solutions; as a consequence, globally optimal solutions cannot be guaranteed in KEV or KEMLLR.[3]

It is worth noting that another nonlinear extension of MLLR using kernel ridge regression (KRR) was proposed by Saon [21] (in the same conference that MPLKR was first published [17]), which we will refer to as KRR-MLLR. There are some subtle differences between MPLKR and KRR-MLLR adaptation method, and we will discuss them in Section III after we have presented MPLKR in details. Moreover, MPLKR can be treated as a kernel version of MAPLR.

The rest of this paper is organized as follows. We first give a review of the ML and MLLR solutions for adaptation in Section II in order to illustrate the regression concept of MLLR. Section III extends linear regression in MLLR to kernel regression in our proposed MPLKR. The section ends with a discussion on the differences between MPLKR and other MLLR variants. Experimental evaluation of MPLKR is presented in Section IV, followed by the concluding remarks and future directions in Section V.

[2]A preliminary version of this paper had been presented in a conference paper [17].

[3]Although our preliminary experiments of fast supervised adaptation on RM in [17] showed that KEMLLR and MPLKR gave comparable performance, KEMLLR ran much more slowly. Thus, KEMLLR it is not further compared with MPLKR in this paper.

II. Adaptation by Maximum-Likelihood Estimation and Linear Regression

Let us consider an SI speech recognition system that employs hidden Markov models (HMMs) with Gaussian mixture states. Assume that there are a total of N Gaussian pdf's, $\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_j, \mathbf{C}_j)$, $j = 1, \ldots, N$, where $\mathbf{o}_t$ is the acoustic vector observed at time t; $\boldsymbol{\mu}_j$ and $\mathbf{C}_j$ are the mean and covariance of the jth Gaussian, respectively.[4] Moreover, the dimension of each acoustic vector is d so that $\mathbf{o}_t \in \mathbb{R}^d$, $\boldsymbol{\mu}_j \in \mathbb{R}^d$, and $\mathbf{C}_j \in \mathbb{R}^{d \times d}$, $j = 1, \ldots, N$. Also assume that there are T speech frames available for adaptation. Only the Gaussian means are adapted, and the new adapted means are denoted by $\hat{\boldsymbol{\mu}}_j$, $j = 1, \ldots, N$.

A. Adaptation by Maximum-Likelihood Re-Estimation (MLRE)

If T is large, meaning that there are lots of adaptation data, one may simply replace the SI Gaussian means by the corresponding ML estimates computed from the adaptation data. That is, the ML mean vectors $\boldsymbol{\mu}_j^* \in \mathbb{R}^d$, $j = 1, \ldots, N$, of the new speaker can be found by maximizing the log-likelihood of the adaptation data (after removing the irrelevant terms) as follows:

$$\boldsymbol{\mu}_j^* = \arg\max_{\hat{\boldsymbol{\mu}}_j} \left[-\sum_{j=1}^{N} \sum_{t=1}^{T} \gamma_j(t) (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j)' \mathbf{C}_j^{-1} (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_j) \right] \tag{1}$$

where $\gamma_j(t)$ is the posterior probability of the jth Gaussian at time t given the T adaptation observations $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$. It can be easily shown that the solution of (1) is

$$\boldsymbol{\mu}_j^* = \frac{\sum_{t=1}^{T} \gamma_j(t) \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_j(t)} = \frac{\mathbf{O}\boldsymbol{\gamma}_j}{\mathbf{1}'\boldsymbol{\gamma}_j} \tag{2}$$

where $\boldsymbol{\gamma}_j = [\gamma_j(1), \ldots, \gamma_j(T)]' \in \mathbb{R}^T$ and $\mathbf{1} = [1, \ldots, 1]' \in \mathbb{R}^T$.

In practice, however, it is difficult to collect sufficient speech covering all phonetic contexts from a new speaker to compute his/her ML Gaussian means reliably. As a consequence, the ML solution of (2) results in a poorly adapted model, producing poor recognition performance. This is particularly true in the case of fast speaker adaptation when there are fewer than 10 s of adaptation speech.

B. Adaptation by Linear Regression

From the above discussion, we see that overfitting with the ML estimated means is not desirable, and some constraints should be imposed in the ML estimation process. For adaptation using linear transformation, the adapted means are constrained to be a linear transformation of the corresponding means. Without loss of generality, the foregoing discussion only deals with finding a global affine transform which is shared by all the N speaker-independent Gaussian means.[5] Thus, for the jth Gaussian, we first augment the mean vector

[4]Matrices and vectors are bold, written in upper case and lower case, respectively. Scalar quantities, including vector or matrix elements, are not bold.

[5]In practice, the SI Gaussian means are generally clustered in a regression tree so that Gaussian means in the same tree node—also known as a regression class—will share the same affine transform. Our description on global transformation can be then applied to each regression class, and all the sufficient statistics should be collected over all Gaussians in the same regression class.

$\boldsymbol{\mu}_j$ to $\boldsymbol{\xi}_j = [\boldsymbol{\mu}_j', 1]' \in \mathbb{R}^{(d+1)}$, and if $\mathbf{W} \in \mathbb{R}^{d\times(d+1)}$ is the required affine transform, then we require the adapted mean $\hat{\boldsymbol{\mu}}_j$ to be $\mathbf{W}\boldsymbol{\xi}_j$. From the regression perspective, the required linear affine transform is the linear regression function that relates the SI means and the adaptation data.

1) Maximum-Likelihood Linear Regression (MLLR): In MLLR [10], the affine transform $\mathbf{W}$ is found by maximizing the likelihood of the adaptation data, or equivalently, the following function:

$$\max_{\mathbf{W}} \left[-\sum_{j=1}^{N}\sum_{t=1}^{T} \gamma_j(t)(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi}_j)'\mathbf{C}_j^{-1}(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi}_j) \right]. \quad (3)$$

The analytic solution for the general case with full Gaussian covariances can be found in [22]. On the other hand, since most HMM-based recognition systems use diagonal Gaussian covariances, there is a simpler solution which solves the transform row by row as follows. Let $\mathbf{w}_i \in \mathbb{R}^{(d+1)}$, $i = 1, \ldots, d$ be the row vectors of $\mathbf{W}$ so that $\mathbf{W}' = [\mathbf{w}_1, \ldots, \mathbf{w}_d]$. Then, we have

$$\mathbf{w}_i = (\Xi\boldsymbol{\Lambda}_i\Xi')^{-1}\Xi\mathbf{z}_i \quad (4)$$

where

$$\begin{aligned} \Xi &= [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N] \in \mathbb{R}^{(d+1)\times N} \\ \boldsymbol{\Lambda}_i &= \operatorname{diag}\left(C_{1i}^{-1}\mathbf{1}'\boldsymbol{\gamma}_1, \ldots, C_{Ni}^{-1}\mathbf{1}'\boldsymbol{\gamma}_N\right) \in \mathbb{R}^{N\times N} \\ \mathbf{z}_i &= \left[C_{1i}^{-1}\sum_{t=1}^{T} o_{ti}\gamma_1(t), \ldots, C_{Ni}^{-1}\sum_{t=1}^{T} o_{ti}\gamma_N(t) \right]' \in \mathbb{R}^N \end{aligned} \quad (5)$$

and C_{ji} is the ith diagonal element of the jth covariance, and o_{ti} is the ith dimension of the acoustic vector $\mathbf{o}_t$.

The computation of a row of the MLLR transform mainly involves an inversion of the $(d+1)\times(d+1)$ matrix $\Xi\boldsymbol{\Lambda}_i\Xi'$. Thus, the computational complexity of MLLR[6] is $O(d^4)$.

2) Least Squares Linear Regression (LSLR): A special case of MLLR was worked out earlier by Hewett [23] by making two assumptions: firstly, all Gaussians (in the same regression class) have the same covariance; second, there is only one alignment—the Viterbi alignment—between the Gaussians and the adapting acoustic vectors.[7] As a result, the problem is reduced to a least squares linear regression, and the solution is given by

$$\mathbf{W} = \mathbf{O}\Xi'(\Xi\Xi')^{-1}. \quad (6)$$

Due to the simplifying assumptions, the LSLR method involves only one inversion of the, again, $(d+1)\times(d+1)$ matrix $\Xi\Xi'$. Thus, its computational complexity is only $O(d^3)$—and is d times faster than MLLR.

III. Maximum Penalized Likelihood Kernel Regression Adaptation (MPLKR)

Let us look at the two ML-based cost functions used by MLRE [(1)] and MLLR [(3)] more closely. One can see that the quantity $\mathbf{W}\boldsymbol{\xi}_j$ of (3) plays the role of $\hat{\boldsymbol{\mu}}_j$ of (1). As the optimal solution of $\hat{\boldsymbol{\mu}}_j$ of (1) is given by $\boldsymbol{\mu}_j^*$ in (2), mean adaptation by MLLR[8] is, in effect, trying to learn a transform $\mathbf{W}$ such that

$$\mathbf{W}[\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N] = [\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_N^*]$$

or

$$\mathbf{W}\Xi = \mathbf{U}^* \quad (7)$$

where $\mathbf{U} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N] \in \mathbb{R}^{d\times N}$ is the collection of the N speaker-independent Gaussian means, and $\mathbf{U}^*$ is the solution given by MLRE adaptation of $\mathbf{U}$. In the linear system represented by (7), there are $d\times(d+1)$ variables in the affine MLLR transform $\mathbf{W}$, and a total of dN constraints provided by the N maximum-likelihood means $\boldsymbol{\mu}_j^*$, $j = 1, \ldots, N$. When $d+1 \geq N$, in general, an exact solution can be found for $\mathbf{W}$ unless the system is inconsistent;[9] in fact, multiple solutions can be found when $d+1 > N$. Consequently, $\hat{\boldsymbol{\mu}}_j$'s obtained from MLLR are the same as the ML means from MLRE of (2). On the other hand, when $d+1 < N$ (and that is the usual case in MLLR adaptation), the system in (7) is over-constrained—with more constraints than variables—and the solution obtained by MLLR will, in general, be different from that obtained in (2).

A. Nonlinear Regression

For the case where $d+1 < N$ in the linear system represented by (7), one may introduce more variables into the system so that the number of variables is greater than or equal to the number of constraints. As a result, one will be able to get back the optimal Gaussian means $\boldsymbol{\mu}_j^*$, $j = 1, \ldots, N$, in the ML sense. In this paper, this is achieved by promoting the problem to nonlinear regression of the ML means. We further apply kernel methods so that the nonlinear regression still will be represented by a linear system similar to the one in (7) but in the kernel-induced feature space. As a consequence, the resulting linear system can be easily solved with an analytic solution that is globally optimal.

However, as we point out in Section II, these ML adapted means are undesirable in fast adaptation because they are poor estimates when the amount of adaptation speech is scarce. The robustness problem will be solved with the use of an appropriate regularization in Section III-D.

B. Empirical Kernel Map

Let us convert the linear system of (7) to another one such that the number of variables is the same as the number of constraints by mapping $\boldsymbol{\xi}_j \in \mathbb{R}^{d+1}$ to $\varphi(\boldsymbol{\xi}_j) \in \mathbb{R}^N$, $j = 1, \ldots, N$. The linear system in (7) then becomes

$$\tilde{\mathbf{W}}\left[\varphi(\boldsymbol{\xi}_j), \ldots, \varphi(\boldsymbol{\xi}_N)\right] = [\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_N^*]$$

or

$$\tilde{\mathbf{W}}\boldsymbol{\Phi} = \mathbf{U}^* \quad (8)$$

[6]Here we assume the use of the Gaussian elimination method for inverting an $n\times n$ matrix which has a computational time complexity of $O(n^3)$. There are faster but more complicated matrix inversion algorithms such as the Strassen inversion algorithm or Coppersmith–Winograd algorithm; it has been also proved that the lower bound for matrix inversion complexity is $O(n^2 \ln n)$.

[7]Consequently, $N = T$ in this special case. That is, each frame is aligned with one of the HMM Gaussians.

[8]All discussions of adaptation in this paper deal with Gaussian means only; other HMM parameters are not modified.

[9]From (7), the system will be inconsistent if for some m and n, $\boldsymbol{\xi}_m = \boldsymbol{\xi}_n$ but $\boldsymbol{\mu}_m^* \neq \boldsymbol{\mu}_n^*$. For an inconsistent system, a least-squares error solution can still be found by using the pseudoinverse.

where $\boldsymbol{\Phi} \equiv [\varphi(\boldsymbol{\xi}_1), \ldots, \varphi(\boldsymbol{\xi}_N)] \in \mathbb{R}^{N \times N}$ represents the collection of φ-mapped augmented Gaussian means, and $\tilde{\mathbf{W}} \in \mathbb{R}^{d \times N}$ is the new transform for vectors in the new N-dimensional feature space introduced by the φ-mapping. One commonly used φ function for finite-dimensional mapping in kernel methods is the *empirical kernel map* [24] defined as follows. For a given set of N speaker-independent Gaussian means $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N\}$ (where $\forall j, \boldsymbol{\xi}_j \in \mathbb{R}^{d+1}$), the empirical kernel map φ is given by

$$\varphi(\cdot) = [k(\boldsymbol{\xi}_1, \cdot), \ldots, k(\boldsymbol{\xi}_N, \cdot)]' \tag{9}$$

where k is a kernel function. Applying the empirical kernel map to $\boldsymbol{\Phi}$, we obtain

$$\boldsymbol{\Phi} = \begin{bmatrix} k(\boldsymbol{\xi}_1, \boldsymbol{\xi}_1) & \cdots & k(\boldsymbol{\xi}_1, \boldsymbol{\xi}_N) \\ \vdots & \cdots & \vdots \\ k(\boldsymbol{\xi}_N, \boldsymbol{\xi}_1) & \cdots & k(\boldsymbol{\xi}_N, \boldsymbol{\xi}_N) \end{bmatrix} \equiv \mathbf{K} \tag{10}$$

where $\mathbf{K}$ is usually called the *kernel matrix*. Hence, (8) can be rewritten as

$$\tilde{\mathbf{W}}\mathbf{K} = \mathbf{U}^*. \tag{11}$$

From [24], we know that when a positive definite kernel (such as the Gaussian kernel) is used, the kernel matrix in (10) is always full rank if all Gaussian means $\boldsymbol{\xi}_j$, $j = 1, \ldots, N$ are distinct.

C. Trivial Solution

In general, the least squares solution of the linear system in (11) can be obtained by minimizing the following Frobenius norm:

$$||\tilde{\mathbf{W}}\mathbf{K} - \mathbf{U}^*||_F^2 \equiv \operatorname{tr}\left[(\tilde{\mathbf{W}}\mathbf{K} - \mathbf{U}^*)'(\tilde{\mathbf{W}}\mathbf{K} - \mathbf{U}^*)\right] \tag{12}$$

and its general solution is

$$\tilde{\mathbf{W}} = \mathbf{U}^*\mathbf{K}^+ \tag{13}$$

where $\mathbf{K}^+ = \mathbf{K}'(\mathbf{K}\mathbf{K}')^{-1}$ is the pseudoinverse of the kernel matrix $\mathbf{K}$. Moreover, if $\mathbf{K}$ is invertible—as in the case when a positive definite kernel is used—the solution is simply given by

$$\tilde{\mathbf{W}} = \mathbf{U}^*\mathbf{K}^{-1}. \tag{14}$$

The nonlinear regression solution of (14) implies that the new adapted means will be exactly equal to the ML means $\mathbf{U}^*$ obtained in (2). In other words, although the transform matrix $\tilde{\mathbf{W}}$ is tied across all Gaussians, unlike MLLR, the use of kernel methods allows the ML means to be perfectly recovered.

D. Regularization

From the regression perspective, linear regression used by MLLR can only capture linear characteristics in the data; on the other hand, nonlinear regression can be overly flexible, attaining zero training error (which is analogous to our situation here where the ML means can be perfectly recovered) and suffers from overfitting. Hence, proper regularization is needed in the use of (nonlinear) kernel regression for fast adaptation so as to capture possible nonlinearity in the data, and at the same time, effectively control the degree of freedom to avoid overfitting. Assuming that we have some prior knowledge of the expected value of $\tilde{\mathbf{W}}$, a regularization term can be added to the cost function of (12) to penalize those solutions of $\tilde{\mathbf{W}}$ that deviate too much from the expected value. The expected value can be derived from a prior distribution of $\tilde{\mathbf{W}}$. In this paper, we investigate, as in the case of MAPLR, the use of the matrix-variate normal distribution to represent the prior density of $\tilde{\mathbf{W}}$. We then require $\tilde{\mathbf{W}}$ to be close to the mean of the normal prior density, which will be denoted by $\tilde{\mathbf{W}}_0$. Consequently, we arrive at the following minimization function

$$\min_{\tilde{\mathbf{W}}, \beta} ||\tilde{\mathbf{W}}\mathbf{K} - \mathbf{U}^*||_F^2 + \beta||\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0||_F^2 \tag{15}$$

where β is the regularization parameter. We refer to this new method as *maximum penalized likelihood kernel regression* adaptation (MPLKR).

By differentiating (15) w.r.t. $\tilde{\mathbf{W}}$ and setting the result to zero, one can easily show that its general solution is given by

$$\tilde{\mathbf{W}} = (\mathbf{U}^*\mathbf{K}' + \beta\tilde{\mathbf{W}}_0)(\mathbf{K}\mathbf{K}' + \beta\mathbf{I})^{-1}. \tag{16}$$

Furthermore, if the kernel matrix $\mathbf{K}$ is symmetric (which is the case in our paper), the solution can be simplified as

$$\tilde{\mathbf{W}} = (\mathbf{U}^*\mathbf{K} + \beta\tilde{\mathbf{W}}_0)(\mathbf{K}^2 + \beta\mathbf{I})^{-1}. \tag{17}$$

The regularization parameter β can be determined empirically by cross-validation.

Equation (17) also shows that the MPLKR transform can be found analytically, and its computation is dominated by the inversion of the $N \times N$ kernel matrix $\mathbf{K}$. Thus, MPLKR has a computational complexity of $O(N^3)$.

1) Choice of $\tilde{\mathbf{W}}_0$: The choice of $\tilde{\mathbf{W}}_0$ represents a bias of where the $\tilde{\mathbf{W}}$ solution should be. A good fail-safe choice of $\tilde{\mathbf{W}}_0$ is one that will reproduce the original SI Gaussian means $\mathbf{U}$ because when there are not sufficient adaptation data, it is safer to fallback to the original SI model without modifying its parameters.[10] If we denote the MPLKR transform that reproduces the SI means as $\tilde{\mathbf{W}}^{(si)}$, it can be obtained by replacing $\mathbf{U}^*$ by $\mathbf{U}$ in (14). That is

$$\tilde{\mathbf{W}}^{(si)} = \mathbf{U}\mathbf{K}^{-1}. \tag{18}$$

In this paper, we choose $\tilde{\mathbf{W}}_0 = \tilde{\mathbf{W}}^{(si)}$ in all experimental evaluations of MPLKR in Section IV.

2) Generation of New Mean Vectors: In practice, not all Gaussian means are observed in the adaptation speech. This is particularly true for fast adaptation with less than 10 s of speech. Thus, only the observed Gaussians[11] are actually used in the above formulation ((1)–(17)) of MLLR or MPLKR. Any mean vector $\hat{\boldsymbol{\mu}}$ of the new adapted model, regardless of whether it is observed in the adaptation data, can be obtained by first augmenting its corresponding SI mean $\boldsymbol{\mu}$ to $\boldsymbol{\xi} = [\boldsymbol{\mu}', 1]'$, mapping $\boldsymbol{\xi}$ using the empirical kernel map, the φ function in (9) to the

[10] In MLLR, the transform that will reproduce the SI means is simply the identity matrix.

[11] In general, how to filter the data before regression is an open question. In this paper, we simply keep all the observed Gaussians because of the observation that standard MLLR also keeps all adaptation data for regression and it performs well with the simple strategy.

kernel-induced feature space, and then multiply it with the $\tilde{\mathbf{W}}$ solution given by (17). That is

$$\begin{aligned}\hat{\boldsymbol{\mu}} &= \tilde{\mathbf{W}}\varphi(\boldsymbol{\xi}) \\ &= (\mathbf{U}^*\mathbf{K} + \beta\tilde{\mathbf{W}}_0)(\mathbf{K}^2 + \beta\mathbf{I})^{-1} \\ &\quad \cdot [k(\boldsymbol{\xi}_1, \boldsymbol{\xi}), \ldots, k(\boldsymbol{\xi}_N, \boldsymbol{\xi})]'. \end{aligned} \tag{19}$$

3) Regularized Linear Regression: For the sake of comparison, we also experiment with the simple case when the mapping function in (8) is the simple identity function $\varphi(x) = x$. MPLKR is then reduced to simple least squares linear regression between the augmented mean vectors and the ML-adapted mean vectors with regularization. By replacing the kernel matrix $\mathbf{K}$ of (15) by $\boldsymbol{\Xi}$ of (7), we obtain

$$\min_{\tilde{\mathbf{W}},\beta} \|\tilde{\mathbf{W}}\boldsymbol{\Xi} - \mathbf{U}^*\|_F^2 + \beta\|\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0\|_F^2. \tag{20}$$

We call this method *maximum penalized likelihood linear regression* (MPLLR) since no nonlinear kernels are employed. Notice that MPLLR is different from LSLR of Section II-B2 which regresses the acoustic observations against the Gaussian means.

4) Relationship With Generalized Tikhonov Regularization: The mathematical form of MPLKR can be thought of as a special case of generalized Tikhonov regularization with the use of Frobenius norm and an identity Tikhonov matrix. However, the motivations behind the use of penalized likelihood and regularization are different. Generalized Tikhonov regularization is usually applied to solving ill-posed problems in the classical sense given by Hadamard.[12] In our case, the problem is well-posed and we use penalized likelihood in the sense given by Green [25] not because of stability issue but to avoid overfitting by biasing the solution toward $\tilde{\mathbf{W}}_0$, which is observed to have reasonably good performance.

The formulation of our MPLKR also shares a Bayesian interpretation similar to that of the generalized Tikhonov regularization, which will be given in Section III-F2.

E. Advantages Over Other Kernel-Based Adaptation

Compared to other kernel-based adaptation techniques recently proposed by us, such as kernel eigenvoice (KEV) [26], embedded kernel eigenvoice (eKEV) [27], and kernel eigenspace-based MLLR (KEMLLR) [28], MPLKR has the advantage that the new adapted Gaussian mean vectors can be computed analytically by simply solving a linear system, whereas the other three kernel-based adaptation methods are usually solved by iterative gradient-based algorithms. As no nonlinear optimization is involved, unlike KEV, eKEV, or KEMLLR adaptation, the solution obtained by MPLKR adaptation is always globally optimal.

F. Difference Between MPLKR and Other MLLR Extensions

MPLKR starts from generalizing MLLR using nonlinear regression with the help of kernel methods. It bears certain similarities with some recent MLLR extensions such as MAPLR [12] and KRR-MLLR [21], and yet they are subtly different.

[12]A problem is well-posed if a solution exists which is unique and varies continuously with the data.

TABLE I
COMPUTATIONAL COMPLEXITY OF MLLR AND ITS VARIANTS. (d IS THE DIMENSION OF ACOUSTIC VECTORS, K IS THE NUMBER OF EIGENMATRICES IN EMLLR, N IS THE NUMBER OF GAUSSIANS IN A REGRESSION CLASS, T IS THE NUMBER OF ADAPTATION SPEECH FRAMES ASSOCIATED WITH THE REGRESSION CLASS)

Method	Complexity	Diagonal Cov?	Viterbi?
MLLR	$O(d^4)$	yes	no
LSLR	$O(d^3)$	no	yes
EMLLR	$O(K^3)$	—	no
MAPLR	$O(d^4)$	yes	no
MPLLR	$O(d^3)$	no	no
MPLKR	$O(N^3)$	no	no
KRR-MLLR	$O(dN^3T^3)$	yes	no

Table I summarizes the computational complexity of these adaptation algorithms, and their assumption of diagonal Gaussian covariances and Viterbi alignment. Notice that all these methods have analytical solutions.

1) MPLKR and MLLR/LSLR: As a side effect of the conversion from an over-constrained linear system [(7)] in the usual case of MLLR to a sufficiently constrained linear system [(11)] in MPLKR, the response variable of the regression changes from the observed adaptation data $\mathbf{O}$ in MLLR to the ML Gaussian means $\mathbf{U}^*$ in MPLKR. MPLKR is similar to LSLR in two aspects. a) Both of them employ least squares regression, but MPLKR does that in the kernel-induced feature space. The use of least squared errors instead of maximum likelihood as the cost function leads to their relatively reduced computational complexity. b) The Gaussian covariances do not appear in their formulation; thus, they have the same solution for both diagonal or full Gaussian covariances. On the other hand, LSLR uses Viterbi alignment and makes a hard decision on assigning an adaptation frame to only one single Gaussian; MPLKR is similar to MLLR and makes a soft decision on the frame assignment, which is weighted by its posterior probability.

2) MPLKR and MAPLR: Our MPLLR may be considered as a MAPLR variant (and MPLKR as a kernel version of MAPLR) though it regresses the original Gaussian means with the ML adapted Gaussian means instead of the adaptation observations. The second term in (15) and (20) implements the prior probability of the MPLKR (and MPLLR) transform $\tilde{\mathbf{W}}$.

That is, if we assume, like MAPLR [13], that the prior density of $\tilde{\mathbf{W}}$ is the matrix normal distribution

$$\begin{aligned} p(\tilde{\mathbf{W}}|\tilde{\mathbf{W}}_0, \boldsymbol{\Omega}, \boldsymbol{\Sigma}) &= (2\pi)^{-dN/2}|\boldsymbol{\Omega}|^{N/2}|\boldsymbol{\Sigma}|^{-d/2} \\ &\quad \times \exp\left(-\frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Omega}^{-1}(\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0)'\boldsymbol{\Sigma}^{-1}(\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0)\right]\right) \end{aligned} \tag{21}$$

where $\tilde{\mathbf{W}}_0 \in \mathbb{R}^{d\times N}$, $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ are the hyperparameters of the distribution, and set $\boldsymbol{\Omega} = \boldsymbol{\Sigma} = \mathbf{I}$, then we have

$$\begin{aligned} &\log p(\tilde{\mathbf{W}}|\tilde{\mathbf{W}}_0, \mathbf{I}, \mathbf{I}) \\ &\quad = \text{constant} - \frac{1}{2}\mathrm{tr}\left[(\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0)'(\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0)\right] \\ &\quad = \text{constant} - \frac{1}{2}\|\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0\|_F^2. \end{aligned} \tag{22}$$

Thus, maximizing the prior probability in MAPLR is equivalent to minimizing the Frobenius norm of $(\tilde{\mathbf{W}} - \tilde{\mathbf{W}}_0)$ in MPLKR (and MPLLR).

However, we prefer using the term "penalizing likelihood" as we are then free to choose a regularizer as we see fit according to our problem without restricting it to be a true prior.

Furthermore, if one denotes the solution that perfectly recovers the ML means shown in (14) as

$$\tilde{\mathbf{W}}^* = \mathbf{U}^*\mathbf{K}^{-1} \tag{23}$$

then (17) can further be written as

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{W}}^*\mathbf{K}^2 + \beta\tilde{\mathbf{W}}_0)(\mathbf{K}^2 + \beta\mathbf{I})^{-1}. \tag{24}$$

Equation (24) shows that the MPLKR solution is similar to the MAP adaptation of Gaussian means using conjugate Gaussian prior: $\tilde{\mathbf{W}}^*$ represents the transform estimated solely from adaptation speech while $\tilde{\mathbf{W}}_0$ is the prior mean, and $\beta\mathbf{I}$ and $\mathbf{K}^2$ control the balance between the contributions of the prior and the adaptation speech in determining the MPLKR transform. Another way to see this is to replace $\mathbf{K}^2$ in (24) by the identity matrix $\mathbf{I}$, then it becomes

$$\tilde{\mathbf{W}} = \frac{\tilde{\mathbf{W}}^* + \beta\tilde{\mathbf{W}}_0}{1+\beta} \tag{25}$$

which is the familiar MAP adaptation solution.

3) MPLKR and KRR-MLLR: Saon's KRR-MLLR adaptation [21] is most similar to our MPLKR: both algorithms are kernel versions of MLLR. Similar to MLLR, Saon applies weighted kernel ridge regression for each dimension of the acoustic vectors and Gaussian means, resulting in the following minimization function [21, Eqn. (15)]) to compute each row of the KRR-MLLR transform

$$\min_{\mathbf{c}}(\mathbf{y} - \mathbf{K}\mathbf{c})'\mathbf{W}(\mathbf{y} - \mathbf{K}\mathbf{c}) + \lambda\mathbf{c}'\mathbf{K}\mathbf{c} \tag{26}$$

where $\mathbf{y} \in \mathbb{R}^{NT}$ represents a particular dimension of all adaptation acoustic vectors; $\mathbf{K} \in \mathbb{R}^{NT\times NT}$ is the kernel matrix; $\mathbf{c} \in \mathbb{R}^{NT}$ is the minimization variable, which, together with the kernel matrix, will be used to compute the new adapted means; $\mathbf{W}$ is a weighting matrix consisting of Gaussian posterior probabilities and Gaussian variances (and is not the MLLR transform); and λ is a regularization parameter. KRR-MLLR has the nice property that if linear kernel function is used and λ is set to zero, it is reduced to MLLR. On the other hand, that does not apply to MPLKR. Nevertheless, we believe that our MPLKR may have two advantages over KRR-MLLR.

1) Since KRR-MLLR follows the formulation development of MLLR very closely except for the use of KRR, its computation complexity is $O(dN^3T^3)$, which is very high. The high complexity is mainly due to the inversion of the $NT \times NT$ kernel matrix, which is T times bigger than ours. [21] suggests two heuristics to reduce the complexity. a) Reduce T by choosing a subset of adaptation frames, or by clustering the data and using only the centroids of the clusters as training data. b) Reduce N by keeping only those Gaussians with posterior probability greater than a threshold. In contrast, the kernel matrix in our MPLKR is much smaller. Thus, MPLKR can solve bigger adaptation problems than KRR-MLLR without resorting to approximation heuristics; when the problem is too big, MPLKR may take advantage of similar approximations as well.
2) The regularizer used in KRR-MLLR is the standard regularizer commonly used in KRR that is not particularly chosen for the current task of speech adaptation. On the other hand, our MPLKR allows the incorporation of informative prior knowledge in the regularizer. We believe that more informative prior should lead to better performance in MPLKR.

IV. Experimental Evaluation

The proposed *maximum penalized likelihood kernel regression* adaptation (MPLKR) was evaluated on speaker adaptation of two continuous speech corpora: the 1000-word DARPA RM [29], and the 5000-word WSJ0 [30]. We first performed *supervised* adaptation on the simpler RM task using context-independent acoustic models to study the behavior of various adaptation methods (especially KRR-MLLR and MPLKR). The simpler task also allows us to run many experiments within reasonable amount of time to investigate the proper settings of various parameters—such as the parameters in the kernel function, and the value of the regularization parameter—in these methods. Afterwards, these system parameters were applied *without any changes* to *unsupervised* speaker adaptation on WSJ0 using cross-word context-dependent acoustic models. Specifically, MPLKR was compared with the following model and adaptation methods.

SI: the baseline SI model.

MAP: the speaker-adapted (SA) model found by MAP adaptation [9].

MLLR/MLLR-B/MLLR-D: the SA model found by MLLR adaptation [10] using full, block-diagonal, or diagonal transform respectively. MLLR-B uses the common three 13-dimensional blocks.

EMLLR: the SA model found by eigenspace-based MLLR adaptation [8].

MAPLR: the SA model found by MAPLR adaptation [12].

KRR-MLLR: the SA model found by KRR-MLLR adaptation [21].

RSW: the SA model found by *reference speaker weighting* [11].

MPLKR: the SA model found by MPLKR.

MPLLR: the SA model found by MPLLR, which is the degenerative MPLKR method when the identity mapping function is used.

For each adaptation method, we tried to find the best setup for the method so as to obtain its best results for comparison. Both MAP and MLLR adaptation were done using the HTK software; only their basic algorithms were employed. For MAP adaptation, scaling factors in the range of 3–30 were tried. For MLLR adaptation, it was performed with a regression tree of 32 classes; the minimum occupation count for a regression class was adjusted for the three different forms of transformation matrix: full transform, block-diagonal transform, or diagonal transform. The adaptation results with the best setup (scaling factor and minimum occupation count respectively) are reported for MAP and MLLR. Thus, the MAP and MLLR results represent an upper bound for these methods. For EMLLR and

TABLE II
BEST VALUES OF GAUSSIAN KERNEL WIDTH σ AND REGULARIZATION PARAMETER β FOUND FOR MAPLR, MPLKR, AND KRR-MLLR ON RM ADAPTATION

Method	σ	β
MAPLR	—	0.1
MPLKR	0.05	0.1
KRR-MLLR	0.05	0.02

RSW adaptation, the speaker-dependent (SD) models for the training speakers were created by MLLR adaptation using the same 32-class regression tree. For RM adaptation, EMLLR used all eigen-matrices for 10-s adaptation and only half of them for 5-s adaptation; similarly, RSW also used all training speakers as the reference speakers for 10-s adaptation and only half of them for 5-s adaptation. For unsupervised adaptation on WSJ0, they used all eigen-matrices and training speakers as reference speakers, respectively. For the two kernel-based adaptation methods, namely, MPLKR and KRR-MLLR, the following Gaussian kernel was used:

$$k(\mathbf{u},\mathbf{v}) = \exp\left(-\sigma\|\mathbf{u}-\mathbf{v}\|^2\right) \tag{27}$$

where σ controls the width of the Gaussian kernel, and $|\mathbf{u}-\mathbf{v}|^2 = (\mathbf{u}-\mathbf{v})'(\mathbf{u}-\mathbf{v})$ is the Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$. To reduce the computation of KRR-MLLR, we followed [21] and used the rectangular approximation method as well as ignored those pairs of adaptation frames and Gaussian means which have a posterior probability of less than 0.1.

Lastly, during supervised adaptation, the contents of the adaptation utterances are assumed to be known and one knows which models to adapt. The SI model was used to compute the initial Gaussian mixture posteriors. In subsequent adaptation iterations, these Gaussian posteriors were estimated using the new adapted model found at the previous iteration. The model obtained from the last adaptation iteration was then used to recognize the test utterances which are different from the adaptation utterances. Unsupervised adaptation ran similarly except 1) since the contents were not known, they were first estimated by Viterbi decoding using the SI model, and 2) each test utterance was its own adaptation source. Thus, after the final adapted model was created, it was used to recognize the same utterance and the recognition accuracy was noted.

A. Supervised Adaptation on RM

1) RM Corpus: The Resource Management corpus RM1 consists of clean read speech that represents queries about the naval resources. The utterances were recorded using a headset microphone at 16 kHz with 16-bit resolution. The corpus comprises a SI section and a SD section. The SI section consists of 3990 training utterances from 109 speakers, whereas there are 12 speakers in the SD section, each having 600 utterances for training, 100 utterances for development, and 100 utterances for evaluation. The corpus has a vocabulary size of 1000 words.

2) Feature Extraction and Acoustic Modeling: All training and testing data were processed to extract 12 static mel-frequency cepstral coefficients (MFCCs) and the normalized frame energy from each speech frame of 25 ms at every 10 ms. Thus, the dimension of acoustic vectors in RM1 is $d = 13$. Forty-seven context-independent and SI phoneme models were trained using only the acoustic observations from the SI training set. Each phoneme model is a strictly left-to-right, three-state hidden Markov model (HMM) with a mixture of ten Gaussian components per state. All Gaussians have diagonal covariances. In addition, there are a three-state "sil" model to capture silence and a 1-state "sp" model to capture short pauses.

TABLE III
PERFORMANCE OF MPLKR ON RM DEVELOPMENT DATA WITH DIFFERENT VALUES OF GAUSSIAN KERNEL WIDTH σ AND REGULARIZATION PARAMETER β

β/σ	0.001	0.005	0.01	0.05	0.1
0.001	64.49	78.60	83.45	79.58	74.90
0.005	65.26	77.21	83.40	81.57	76.73
0.01	64.80	77.28	83.37	81.93	77.64
0.05	62.28	78.29	83.06	83.47	79.89
0.1	66.27	77.88	83.45	**83.78**	80.61
0.5	62.98	77.74	83.42	83.35	82.58
1	61.97	77.59	83.69	83.25	82.82

3) Experimental Procedure: Adaptation was performed using 5 s and 10 s of speech data (or, about 4.6 s and 9.2 s if we exclude the leading and ending silence) among the 100 development utterances of each test speaker. The adapted models were then tested on the 100 evaluation utterances of their speakers using the standard RM word-pair grammar which has a perplexity of 60. To improve the statistical reliability of the results, for each test speaker, three sets of adaptation data of the required duration were randomly chosen from his/her development utterances. Reported results are the averages of experiments over the three adaptation sets of the 12 test speakers.

For each method, three adaptation iterations were run. The system parameters, the Gaussian kernel width σ and the regularization parameter β for MAPLR, MPLKR, and KRR-MLLR were determined as follows: from the 100 development utterances of each speaker that had not been selected for adaptation, 40 utterances were chosen to tune σ and β using a grid search. The set of σ and β that gives the best average performance over all 12 speakers was then adopted. Table II lists their best values for the three methods, and Table III shows the sensitivity of MPLKR on the values of α and β. It is interesting to see that the two MAP-based methods (MAPLR and MPLKR) use the same regularization parameter for their priors, and KRR-MLLR requires a much smaller contribution from its prior.

4) Adaptation Performance Comparison: Using the best values of σ and β in Table II, MPLKR was compared with other MLLR variants. The comparison results are summarized in Table IV. The bold figures represent the best results in each column. We observe the following.

- When there are only 5 s of adaptation speech, only MAPLR, EMLLR, and RSW are effective; but the simple EMLLR and RSW are much better.
- When there are 10 s of speech, all adaptation methods except MAP work reasonably well. In addition:
  - — MLLR with full transform lives up to the expectation as being one of the best adaptation methods when there are 10 or more seconds of adaptation speech.
  - — MPLKR and RSW work best, achieving basically the same performance that is slightly better than MLLR.

TABLE IV
PERFORMANCE OF SUPERVISED ADAPTATION ON RM. RESULTS ARE WORD ACCURACIES IN % ON THE TEST SET

Model/Method	5s	10s
SI	78.27	78.27
MAP	78.41	79.82
MLLR	78.43	83.26
EMLLR	**80.96**	82.68
RSW	80.46	**83.42**
MAPLR	79.81	82.01
KRR-MLLR	78.42	81.60
KRR-MLLR (linear)	74.85	82.34
MPLLR	78.27	82.76
MPLKR	78.64	**83.40**

— KRR-MLLR does not perform as well as we expect. We re-ran using linear kernel and got more reasonable performance (which is labeled as "KRR-MLLR (linear)" in Table IV) though it is still short of MLLR's performance.[13]
— The simple MPLLR (MPLKR with linear mapping) works surprisingly well with 10 s but poorly with 5 s. Its poor performance at 5 s is not unexpected as the ML means estimated with 5 s of speech are probably too bad for subsequent linear regression. On the contrary, MAPLR, which is similar to MPLLR but regresses directly with the adaptation observations, performs much better in the 5-s adaptation task.
— The better performance of MPLKR over MPLLR is attributed to the exploitation of nonlinearity in their framework with the use of a kernel map.

In summary, the adaptation experiments in this simple RM adaptation task show that when there are less than 5 s of adaptation speech, one should choose simple methods such as EMLLR or RSW that exploit correlation among the acoustic units; on the other hand, when there are about 10 s of adaptation data, more sophisticated adaptation methods, such as MPLKR, start to pay off although the simple RSW method performs unexpectedly well in both 5 s and 10 s adaptation. Table V shows the results of four common significance tests; they confirm that on the 10 s RM supervised adaptation task, MPLKR performs significantly better than all other methods in Table IV except RSW at the 95% confidence level.

B. Evaluation on Large-Vocabulary Continuous Speech Recognition (LVCSR)

In this section, experience we learned from supervised adaptation of the simpler context-independent RM task is used to check if MPLKR adaptation is also effective for unsupervised adaptation on a relatively large-vocabulary recognition task using context-dependent HMMs.

1) WSJ0 Corpus: The Wall Street Journal corpus WSJ0 [30] with 5K vocabulary was chosen. The standard SI-84 training set was used for training the SI model. It consists of 83 speakers and 7138 utterances for a total of about 14 h of training speech (after discarding the problematic data from one speaker). The standard nov'92 5K non-verbalized test set was used for evaluation. It consists of eight speakers, each with about 40 utterances.

[13]Theoretically, KRR-MLLR with linear kernel is equivalent to MLLR. However, in practice, due to the use of heuristics in KRR-MLLR to reduce its computation, they differ.

TABLE V
SIGNIFICANCE TEST RESULTS AT THE 95% CONFIDENCE LEVEL FOR THE COMPARISON BETWEEN MPLKR AND OTHER ADAPTATION METHODS FOR THE 10 S RM ADAPTATION TASK. (A "√" MEANS THAT MPLKR IS SIGNIFICANTLY BETTER, AND A '×' MEANS POORER. THE SIGNIFICANCE TESTS ARE: MP = MATCHED PAIR TEST, SI = SIGNED PAIRED COMPARISON TEST, WI = WILCOXON SIGNED RANK TEST, MN = MCNEMAR TEST)

Method	MP	SI	WI	MN
SI	√	√	√	√
MAP	√	√	√	√
MLLR	√	√	√	√
EMLLR	√	√	√	√
RSW	×	same	same	×
MAPLR	√	√	√	√
KRR-MLLR	√	√	√	√
KRR-MLLR (linear)	√	√	√	√
MPLLR	√	√	√	√

2) Feature Extraction and Acoustic Modeling: The traditional 39-dimensional MFCC vectors were extracted at every 10 ms over a window of 25 ms from the training and testing data. The SI model consists of 15 449 cross-word triphones based on 39 base phonemes. Each triphone was modeled as a continuous density HMM which is strictly left-to-right and has three states with a Gaussian mixture density of 16 components per state. State tying was performed to give 3131 tied states in the final SI model. In addition, the same type of "sil" and "sp" models were trained as in the last RM evaluation.

3) Experimental Procedure: For each speaker, unsupervised adaptation was performed using one of his 40 utterances at a time. Then the adapted model was used to decode the same utterance. Reported results are based on the average over all 40 utterances of all eight speakers (i.e., totally 320 test utterances). Each adaptation method was run for six iterations to study their convergence behavior. Notice that the average length of each WSJ0 utterance is 7.26 s (or, 6 s if one excludes the silence portions). The system parameters such as the Gaussian kernel width σ and regularization parameter β for various methods were simply adopted from the corresponding values found in RM adaptation.

4) Adaptation Performance Comparison: Table VI summarizes the performance of the various unsupervised adaptation methods on WSJ0 at the end of each adaptation iteration. The best result for each method is bold, and if two iterations have the same performance, the one occurring with fewer iterations is highlighted.

The results are similar to those of 10-s RM adaptation. Nevertheless, probably because WSJ0 is a much more difficult recognition task with many more Gaussian parameters than RM, there are also some differences. The most notable difference is that now all MLLR variants seem to work in this task. KRR-MLLR now performs as well as MLLR, but MPLKR and RSW continue to outperform all other methods. The results of four common significance tests in Table VII once again confirm that on the WSJ unsupervised adaptation task, MPLKR performs significantly better than all other methods in Table VI except RSW

TABLE VI
PERFORMANCE OF UNSUPERVISED ADAPTATION ON WSJ. RESULTS ARE WORD ACCURACIES IN % ON THE TEST SET

Model or Method	Iteration					
	1	2	3	4	5	6
SI	**92.60**	92.60	92.60	92.60	92.60	92.60
MAP	**92.60**	92.60	92.60	92.60	92.60	92.60
MLLR	92.75	92.83	**92.85**	92.85	92.85	92.85
MLLR-D	93.05	93.16	93.16	**93.20**	93.20	93.20
MLLR-B	93.16	93.20	**93.24**	93.18	93.18	93.24
EMLLR	93.37	**93.61**	93.52	93.52	93.54	93.49
RSW	93.48	93.56	93.67	93.69	93.67	**93.70**
MPLLR	92.69	92.75	**92.82**	92.82	92.82	92.82
MAPLR	**93.05**	92.96	92.94	92.96	92.81	92.87
KRR-MLLR	92.87	92.98	93.12	93.23	**93.28**	93.27
MPLKR	93.23	93.37	93.54	93.62	93.73	**93.75**

TABLE VII
SIGNIFICANCE TEST RESULTS AT THE 95% CONFIDENCE LEVEL FOR THE COMPARISON BETWEEN MPLKR AND OTHER ADAPTATION METHODS FOR THE WSJ ADAPTATION TASK. (A "√" MEANS THAT MPLKR IS SIGNIFICANTLY BETTER. THE SIGNIFICANCE TESTS ARE: MP = MATCHED PAIR TEST, SI = SIGNED PAIRED COMPARISON TEST, WI = WILCOXON SIGNED RANK TEST, MN = MCNEMAR TEST)

Method	MP	SI	WI	MN
SI	√	√	√	√
MAP	√	√	√	√
MLLR-B	√	√	√	√
EMLLR	√	√	same	√
RSW	√	same	same	√
MAPLR	√	√	√	√
KRR-MLLR	√	√	√	√
MPLLR	√	√	√	√

at the 95% confidence level, and RSW is probably comparable with MPLKR.

On the other hand, the various methods have very different convergence behavior. To see that, the adaptation performance of MLLR-B, RSW, MAPLR, KRR-MLLR, and MPLKR across iterations is replotted in Fig. 1. It is noticed that MLLR-B does not change much with iterations, while KRR-MLLR, RSW, and MPLKR do. On the other hand, MAPLR actually performs worse with more iterations. RSW converges faster than the two kernel-based MLLR variants, namely, KRR-MLLR and MPLKR: RSW converges in about three iterations, while KRR-MLLR and MPLKR converge in about six iterations.

In summary, once again, MPLKR and RSW perform the best.

V. CONCLUSION

In this paper, we try to improve the standard MLLR speaker adaptation method by using kernel methods to capture possible nonlinearity in the data under the MLLR framework. Unlike the previous kernel-based adaptation methods (such as KEV, eKEV, and KEMLLR) we proposed, the new method, which we call MPLKR, is computationally simpler and the solution can be analytically obtained by simply solving a linear system. No nonlinear optimization is involved, and the solution obtained by MPLKR is always globally optimal. In both supervised adaptation on the Resource Management task and unsupervised adaptation on the more difficult Wall Street Journal task using

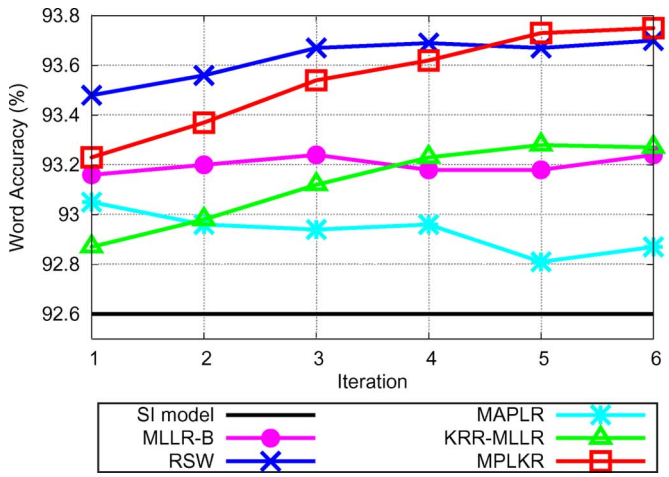


Fig. 1. Convergence behavior of various unsupervised adaptation methods on WSJ.

about 10 s of speech, MPLKR outperformed all other adaptation methods that we tried except RSW which gives comparable performance. For example, for the WSJ0 task, unsupervised adaptation using MPLKR reduces the word error rate of the SI model by 15.5%, whereas the figure for MLLR is 8.65%.

However, we are cautious to see that the simple linear method, RSW, performs as well as the kernel-based MPLKR in both adaptation tasks. In our experience, RSW saturates very fast and does not improve much after 10 s of adaptation speech. On the other hand, MPLKR will not have this limitation: with more adaptation data, the ML means that the method uses for kernel regression will be estimated more reliably, and the subsequent regression will be more accurate.

In summary, we find that RSW and MPLKR perform very well for the fast adaptation tasks in this paper. RSW is simple and very fast, but it saturates quickly after approximately 10 s of adaptation speech. The standard MLLR is well studied and is known to perform reasonably well with a wide range of amount of adaptation data if used together with an appropriate regression class tree of Gaussians. Our new MPLKR is somewhere between RSW and MLLR: it performs as well as RSW in fast adaptation but is slower and has a speed similar to that of MLLR; on the other hand, in theory, it has the potential to work well for longer adaptation data by using a regression class tree as MLLR, and by suitably adjusting the regularizer.

REFERENCES

[1] J. R. Bellegarda, P. V. de Souza, A. J. Nadas, D. Nahamoo, M. A. Picheny, and L. R. Bahl, "Robust speaker adaptation using a piecewise linear acoustic mapping," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, 1992, vol. 1, pp. 445–448.

[2] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, Apr. 1998.

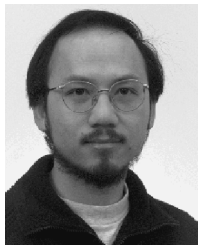
[3] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.

[4] T. Kosaka, S. Matsunaga, and S. Sagayama, "Speaker-independent speech recognition based on tree-structured speaker clustering," *Comput. Speech Lang.*, vol. 10, pp. 55–74, 1996.

[5] M. F. J. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.

[6] T. J. Hazen, "A comparison of novel techniques for rapid speaker adaptation," *Speech Commun.*, vol. 31, pp. 15–33, May 2000.

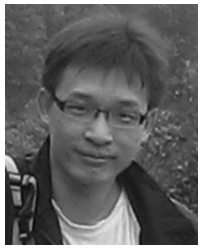
[7] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 695–707, Nov. 2000.
[8] K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihood linear regression," in *Proc. Int. Conf. Spoken Lang. Process.*, 2000, vol. 3, pp. 742–745.
[9] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.
[10] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, pp. 171–185, 1995.
[11] B. Mak, T.-C. Lai, and R. Hsiao, "Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 14–19, 2006, vol. 1, pp. 229–232.
[12] C. Chesta, O. Siohan, and C. H. Lee, "Maximum a posteriori linear regression for hidden Markov model adaptation," in *Proc. Eur. Conf. Speech Commun. Technol.*, 1999, vol. 1, pp. 211–214.
[13] O. Siohan, T. A. Myrvoll, and C. H. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," *Comput. Speech Lang.*, vol. 16, pp. 5–24, Jan. 2002.
[14] K. Shinoda and C. H. Lee, "Unsupervised adaptation using structural Bayes approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1998, pp. 793–796.
[15] H. Erdogan, Y. Q. Gao, and M. Picheny, "Rapid adaptation using penalized-likelihood methods," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 1, pp. 333–336.
[16] A. Gunawardana and W. Byrne, "Discounted likelihood linear regression for rapid speaker adaptation," *Comput. Speech Lang.*, vol. 15, pp. 15–38, Jan. 2001.
[17] I. W. Tsang, J. T. Kwok, B. Mak, K. Zhang, and J. J. Pan, "Fast speaker adaptation via maximum penalized likelihood kernel regression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 14–19, 2006, vol. 1, pp. 997–1000.
[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[19] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
[20] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[21] G. Saon, "A non-linear speaker adaptation technique using kernel ridge regression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, vol. I, pp. 225–228.
[22] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, Apr. 1998.
[23] A. J. Hewett, "Training and Speaker Adaptation in Template-Based Speech Recognition," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.
[24] B. Schölkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
[25] P. J. Green, "Penalized likelihood," in *Encyclopedia of Statistical Sciences, Update Volume 2*. New York: Wiley, 1998, pp. 578–586.
[26] B. Mak, J. T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 984–992, Sep. 2005.
[27] B. Mak, R. Hsiao, S. Ho, and J. T. Kwok, "Embedded kernel eigenvoice speaker adaptation and its implication to reference speaker weighting," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 4, pp. 1267–1280, Jul. 2006.
[28] B. Mak and R. Hsiao, "Kernel eigenspace-based MLLR adaptation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 784–795, Mar. 2007.
[29] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-word resource management database for continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1988, vol. 1, pp. 651–654.
[30] D. B. Paul and J. M. Baker, "The design of the Wall Street Journal-based CSR corpus," in *Proc. DARPA Speech Natural Lang. Workshop*, Feb. 1992.



Brian Kan-Wing Mak (M'02) received the B.Sc. degree in electrical engineering from the University of Hong Kong in 1983, the M.S. degree in computer science from the University of California, Santa Barbara, in 1989, and the Ph.D. degree in computer science from Oregon Graduate Institute of Science and Technology, Portland, in 1998.

From 1990 until 1992, he was a Research Programmer at the Speech Technology Laboratory, Panasonic Technologies, Inc., Santa Barbara. From 1997 until receiving the Ph.D. degree in 1998, he was also a Research Consultant at the AT&T Labs—Research, Florham Park, NJ. Since April 1998, he has been with the Department of Computer Science, Hong Kong University of Science and Technology, and is now an Associate Professor. He had been a Visiting Researcher of Bell Laboratories in Summer 2001 and Advanced Telecommunication Research Institute—International in Spring 2003.

He currently serves on the editorial board of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and the *Speech Communication Journal*. His interests include acoustic modeling, speech recognition, spoken language understanding, computer-assisted language learning, and machine learning. He received the Best Paper Award in the area of Speech Processing from the IEEE Signal Processing Society in 2004.



Tsz-Chung Lai received the B.Eng. degree (first class honors) in computer science (minor in mathematics) from the Hong Kong University of Science and Technology in 2004. He is currently pursuing the M.Phil. degree at the Hong Kong University of Science and Technology.

His research interests include speech recognition, speaker adaptation, and kernel methods.

Ivor W. Tsang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology (HKUST) in 2007.

He is currently an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore. His scientific interests include machine learning, kernel methods, and large-scale optimization, and their applications to data mining and pattern recognitions.

Dr. Tsang was awarded the prestigious IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2006. In 2009, he clinched the second-class prize of the National Natural Science Award 2008, China. He was also awarded the Microsoft Fellowship in 2005, the Best Paper Award from the IEEE Hong Kong Chapter of Signal Processing Postgraduate Forum in 2006, and also the HKUST Honor Outstanding Student in 2001.

James Tin-Yau Kwok (SM'07) received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 1996.

He then joined the Department of Computer Science, Hong Kong Baptist University, as an Assistant Professor. He returned to the Hong Kong University of Science and Technology in 2000 and is now an Associate Professor in the Department of Computer Science and Engineering. His research interests include kernel methods, machine learning, pattern recognition, and artificial neural networks.

Prof. Kwok is currently serving as Associate Editors for the IEEE TRANSACTIONS ON NEURAL NETWORKS and the *Neurocomputing Journal*. He also received the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2006.