Maximum Penalized Likelihood Kernel Regression for Fast Adaptation

Brian Mak, Tsz-Chung Lai, Ivor W. Tsang, and James T. Kwok

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 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 be considered as the kernel version of maximum a posteriori linear regression (MAPLA) adaptation. Supervised and unsupervised speaker adaptation using MLLR were evaluated on the Resource Management and Wall Street Journal 5K task respectively, achieving a word error rate reduction of 23.6% and 15.5% respectively over the speaker-independently model.

Keywords—speaker adaptation, maximum likelihood linear regression, reference speaker weighting, kernel regression

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 (fMLLR) [2], [3], the testing acoustic features are modified to match closer to the training acoustic features. 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], [5], [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 seconds 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 rely on a 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 seconds of adaptation speech, using the solution from cluster weighting adaptation as the prior 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 seconds 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], [19], [20]. The basic idea is to first map the SI mean vectors to a high-dimensional feature space via some nonlinear map \( \varphi \) 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 Gaussian mixture states. Assume that there are a total of \( N \) Gaussian pdf’s, \( \mathcal{N}(\mathbf{o}_t; \mu_j, \Sigma_j) \), \( j = 1, \ldots, N \), where \( \mathbf{o}_t \) is the acoustic vector observed at time \( t \); \( \mu_j \) and \( \Sigma_j \) are the mean and covariance of the \( j \)th Gaussian respectively\(^4\). Moreover, the dimension of each acoustic vector is \( d \) so that \( \mathbf{o}_t \in \mathbb{R}^d \), \( \mu_j \in \mathbb{R}^d \), and \( C_j \in \mathbb{R}^{d \times d}, j = 1, \ldots, N \). Also assume that there are \( T \) speech frames available for adaptation.

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 maximum likelihood (ML) estimates computed from the adaptation data. That is, the ML mean vectors \( \mu_j^* \in \mathbb{R}^d, j = 1, \ldots, N \), of the new speaker can be found by maximizing the log-likelihood of the adaptation data (after removing the irrelevant terms) as follows:

\[
\mu_j^* = \arg \max \mu_j \left[ - \sum_{j=1}^{N} \sum_{t=1}^{T} \gamma_j(t)(\mathbf{o}_t - \hat{\mu}_j)'C_j^{-1}(\mathbf{o}_t - \hat{\mu}_j) \right],
\]

(1)

where \( \gamma_j(t) \) is the posterior probability of the \( j \)th 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 Eqn. (1) is

\[
\mu_j^* = \frac{\sum_{t=1}^{T} \gamma_j(t) \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_j(t)} = \frac{\mathbf{O}_T}{\mathbf{1}^T \gamma_j},
\]

(2)

where \( \gamma_j = [\gamma_j(1), \ldots, \gamma_j(T)]' \in \mathbb{R}^T \) 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 Eqn. (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 10s 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 \( j \)th Gaussian, we first augment the mean vector \( \mu_j \) to \( \xi_j = [\mu_j', 1]' \in \mathbb{R}^{(d+1)} \), and

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

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

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

---

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

3Although 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.
if \( W \in \mathbb{R}^{d \times (d+1)} \) is the required affine transform, then we require the adapted mean \( \mu_j \) to be \( W \xi_j \). From the regression perspective, the required linear affine transform is the linear regression function that relates the SI means and the adaptation data.

### B.1 Maximum Likelihood Linear Regression (MLLR)

In MLLR [10], the affine transform \( W \) is found by maximizing the likelihood of the adaptation data, or equivalently, the following function:

\[
\max_W \left[ -\sum_{j=1}^{N} \sum_{t=1}^{T} \gamma_j(t) (\mathbf{o}_t - W \xi_j)^T C_j^{-1} (\mathbf{o}_t - W \xi_j) \right].
\]

(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 \( w_i \in \mathbb{R}^{d+1} \), \( i = 1, \ldots, d \) be the row vectors of \( W \) so that \( W^T = [w_1, \ldots, w_d] \). Then, we have

\[
w_i = (\Xi \Lambda \Xi')^{-1} \Xi z_i,
\]

(4)

where

\[
\Xi = [\xi_1, \ldots, \xi_N] \in \mathbb{R}^{(d+1) \times N},
\]

\[
\Lambda_i = \text{diag}(C_{1i}^{-1} \gamma_1, \ldots, C_{Ni}^{-1} \gamma_N) \in \mathbb{R}^{N \times N},
\]

\[
z_i = \left[ C_{1i}^{-1} \sum_{t=1}^{T} \alpha_t \gamma_1(t), \ldots, C_{Ni}^{-1} \sum_{t=1}^{T} \alpha_t \gamma_N(t) \right]^T \in \mathbb{R}^N
\]

and \( C_{ji} \) is the \( j \)th diagonal element of the \( j \)th covariance, and \( \alpha_t \) is the \( t \)th 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 \( \Xi \Lambda \Xi' \). Thus, the computational complexity of MLLR\(^6\) is \( O(d^3) \).

### B.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; secondly, 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

\[
W = \Omega \Xi \Xi'^{-1} \Xi \Xi'^{-1}.
\]

(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.

\(^6\)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 \( O(n^\log_2 7) \).

\(^7\)Consequently, \( N = T \) in this special case. That is, each frame is aligned with one of the HMM Gaussians.

### III. Maximum Penalized Likelihood Kernel Regression Adaptation (MPLKR)

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

\[
W[\xi_1, \ldots, \xi_N] = [\mu_1^*, \ldots, \mu_N^*]
\]

or

\[
W \Xi = U^*.
\]

(7)

where \( U = [\mu_1, \ldots, \mu_N] \in \mathbb{R}^{d \times N} \) is the collection of the \( N \) speaker-independent Gaussian means, and \( U^* \) is the solution given by MLRE adaptation of \( U \). In the linear system represented by Eqn. (7), there are \( d \times (d+1) \) variables in the affine MLLR transform \( W \), and a total of \( dN \) constraints provided by the \( N \) maximum-likelihood means \( \mu_j^*, j = 1, \ldots, N \). When \( d + 1 \geq N \), in general, an exact solution can be found for \( W \) unless the system is inconsistent\(^9\); in fact, multiple solutions can be found when \( d + 1 > N \). Consequently, \( \mu_j^* \)‘s obtained from MLLR are the same as the ML means from MLRE of Eqn. (2). On the other hand, when \( d + 1 < N \) (and that is the usual case in MLLR adaptation), the system in Eqn. (7) is over-constrained — with more constraints than variables — and the solution obtained by MLLR will, in general, be different from that obtained in Eqn. (2).

### A. Nonlinear Regression

For the case where \( d + 1 < N \) in the linear system represented by Eqn. (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 \( \mu_j^*, j = 1, \ldots, N \), 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 Eqn. (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’s convert the linear system of Eqn. (7) to another one such that the number of variables is the same as the

\(^8\)All discussions of adaptation in this paper deal with Gaussian means only; other HMM parameters are not modified.

\(^9\)From Eqn. (7), the system will be inconsistent if for some \( m \) and \( n \), \( \xi_m = \xi_n \) but \( \mu_m^* \neq \mu_n^* \). For an inconsistent system, a least-squares error solution can still be found by using the pseudo-inverse.
number of constraints by mapping $\xi_j \in \mathbb{R}^{d+1}$ to $\varphi(\xi_j) \in \mathbb{R}^N$, $j = 1, \ldots, N$. The linear system in Eqn. (7) then becomes

$$\tilde{W}[\varphi(\xi_1), \ldots, \varphi(\xi_N)] = [\mu_1^*, \ldots, \mu_N^*]$$

or

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

where $\Phi \equiv [\varphi(\xi_1), \ldots, \varphi(\xi_N)] \in \mathbb{R}^{N \times N}$ represents the collection of $\varphi$-mapped augmented Gaussian means, and $\tilde{W} \in \mathbb{R}^{d \times N}$ is the new transform for vectors in the new $N$-dimensional feature space introduced by the $\varphi$-mapping. One commonly used $\varphi$ 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 $\{\xi_1, \ldots, \xi_N\}$ (where $\forall j, \xi_j \in \mathbb{R}^{d+1}$), the empirical kernel map $\varphi$ is given by

$$\varphi(\cdot) = [k(\xi_1, \cdot), \ldots, k(\xi_N, \cdot)], \quad (9)$$

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

$$\Phi = \begin{bmatrix} k(\xi_1, \xi_1) & \ldots & k(\xi_1, \xi_N) \\ \vdots & \ddots & \vdots \\ k(\xi_N, \xi_1) & \ldots & k(\xi_N, \xi_N) \end{bmatrix} \equiv K, \quad (10)$$

where $K$ is usually called the kernel matrix. Hence, Eqn. (8) can be rewritten as

$$\tilde{W}K = U^*. \quad (11)$$

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

C. Trivial Solution

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

$$||\tilde{W}K - U^*||^2_F \equiv \text{tr}((\tilde{W}K - U^*)(\tilde{W}K - U^*)) \quad (12)$$

And its general solution is

$$\tilde{W} = U^*K^+, \quad (13)$$

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

$$\tilde{W} = U^*K^{-1}. \quad (14)$$

The nonlinear regression solution of Eqn. (14) implies that the new adapted means will be exactly equal to the ML means $U^*$ obtained in Eqn. (2). In other words, although the transform matrix $\tilde{W}$ 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 $\tilde{W}$, a regularization term can be added to the cost function of Eqn. (12) to penalize those solutions of $\tilde{W}$ that deviate too much from the expected value. The expected value can be derived from a prior distribution of $\tilde{W}$. 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 $\tilde{W}$. We then require $\tilde{W}$ to be close to the mean of the normal prior density, which will be denoted by $W_0$. Consequently, we arrive at the following minimization function

$$\min_{\tilde{W}, \beta} ||\tilde{W}K - U^*||^2_F + \beta||\tilde{W} - W_0||^2_F, \quad (15)$$

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

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

$$\tilde{W} = (U^*K' + \beta W_0)(KK' + \beta I)^{-1}. \quad (16)$$

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

$$\tilde{W} = (U^*K + \beta W_0)(K^2 + \beta I)^{-1}. \quad (17)$$

The regularization parameter $\beta$ can be determined empirically by cross-validation.

Eqn. (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 $K$. Thus, MPLKR has a computational complexity of $O(N^3)$.

D.1 Choice of $W_0$

The choice of $W_0$ represents a bias of where the $\tilde{W}$ solution should be. A good fail-safe choice of $W_0$ is one that will reproduce the original SI Gaussian means $U$ because when there are not sufficient adaptation data, it is safer to fallback to the original SI model without modifying its parameters.\textsuperscript{10} If we denote the MPLKR transform that reproduces the SI means as $\tilde{W}^{(si)}$, it can be obtained by replacing $U^*$ by $U$ in Eqn. (14). That is,

$$\tilde{W}^{(si)} = UK^{-1}. \quad (18)$$

\textsuperscript{10}In MLLR, the transform that will reproduce the SI means is simply the identity matrix.
In this paper, we choose $\hat{W}_0 = \hat{W}^{(s)}$ in all experimental evaluations of MPLKR in Section IV.

D.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 10s of speech. Thus, only the observed Gaussians are actually used in the above formulation (Eqns. (1) – (17)) of MLLR or MPLKR. Any mean vector $\hat{\mu}$ 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 $\mu$ to $\xi = [\hat{\mu}^T, 1]^T$, mapping $\xi$ using the empirical kernel map, the $\varphi$ function in Eqn. (9) to the kernel-induced feature space, and then multiply it with the $\hat{W}$ solution given by Eqn. (17). That is,

$$\hat{\mu} = \hat{W}\varphi(\xi) = \left(U^T K + \beta \hat{W}_0 \right) (K^2 + \beta I)^{-1} .\nonumber$$

\begin{align}
- [k(\xi_1, \xi), \ldots, k(\xi_N, \xi)]^T.
\end{align}

D.3 Regularized Linear Regression

For the sake of comparison, we also experiment with the simple case when the mapping function in Eqn. (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 $K$ of Eqn. (15) by $\Xi$ of Eqn. (7), we obtain

$$\min_{\hat{W}, \beta} ||\hat{W} \Xi - U^*||_F^2 + \beta ||\hat{W} - \hat{W}_0||_F^2. \nonumber$$

\begin{align}
\text{(20)}
\end{align}

We call this method maximum penalized likelihood linear regression (MPLL) since no nonlinear kernels are employed. Notice that MPLLR is different from LSLR of Section II-B.2 which regresses the acoustic observations against the Gaussian means.

D.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. In our case, the problem is well-posed and we use penalized likelihood in the sense given by Green because of stability issue but to avoid overfitting by biasing the solution toward $\hat{W}_0$, which is observed to have reasonably good performance.

\[11\] 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.

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

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-F.2.

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.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
Method & Complexity & Diagonal Cov? & Viterbi? \\
\hline
MLLR & $O(d^3)$ & yes & no \\
LSLR & $O(d^3)$ & no & yes \\
EMLLR & $O(K^3)$ & — & no \\
MAPLR & $O(d^3)$ & yes & no \\
MPLLR & $O(d^3)$ & no & no \\
MPLKR & $O(N^3)$ & no & no \\
KRR-MLLR & $O(dN^3T^4)$ & yes & no \\
\hline
\end{tabular}
\caption{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.)}
\end{table}

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 slightly different. 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.

F.1 MPLKR and MLLR/LSLR

As a side effect of the conversion from an over-constrained linear system (Eq. (7)) in the usual case of MLLR to a sufficiently constrained linear system (Eq. (11)) in MPLKR, the response variable of the regression changes from the observed adaptation data $O$ in MLLR to the ML Gaussian means $U^*$ in MPLKR. MPLKR is similar to LSLR in two aspects: (1) 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. (2) 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.

F.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 Eqn. (15) and Eqn. (20) implements the prior probability of the MPLKR (and MPLLR). The second term in Eqn. (15) and Eqn. (20) implements the prior probability in MAPLR is equivalent to minimizing the Frobenius norm of $\tilde{W}$.

Furthermore, if one denotes the solution that perfectly regress the original Gaussian means with the ML adapted Gaussian means as $\tilde{W}$, then we have

$$p(\tilde{W}|\tilde{W}_0, \Omega, \Sigma) = (2\pi)^{-dN/2}|\Omega|^{-N/2}|\Sigma|^{-d/2} \times 
\exp\left(-\frac{1}{2} \Omega^{-1}(\tilde{W} - \tilde{W}_0)\Sigma^{-1}(\tilde{W} - \tilde{W}_0)^\top\right).$$

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

$$\log p(\tilde{W}|\tilde{W}_0, I, I) = \text{constant} - \frac{1}{2} \text{tr}[(\tilde{W} - \tilde{W}_0)'(\tilde{W} - \tilde{W}_0)].$$

Thus, maximizing the prior probability in MAPLR is equivalent to minimizing the Frobenius norm of $(\tilde{W} - \tilde{W}_0)$ 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 Eqn. (14) as

$$W^* = U'K^{-1},$$

then Eqn. (17) can further be written as

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

Eqn. (24) shows that the MPLKR solution is similar to the MAP adaptation of Gaussian means using conjugate Gaussian prior: $W^*$ represents the transform estimated solely from adaptation speech while $\tilde{W}_0$ is the prior mean, and $\beta$ and $K^2$ 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 $K^2$ in Eqn. (24) by the identity matrix $I$, then it becomes

$$\tilde{W} = \frac{W^* + \beta\tilde{W}_0}{1 + \beta}.$$
experiments within reasonable amount of time to investigate the proper settings of various parameters — such as the 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 speaker-independent (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.
- **MPLLKR**: the SA model found by MPLLKR, 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 RSW adaptation, the 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 10s adaptation and only half of them for 5s adaptation; similarly, RSW also used all training speakers as the reference speakers for 10s adaptation and only half of them for 5s 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:

\[ k(u, v) = \exp(-\sigma||u - v||^2) \]  

where \( \sigma \) controls the width of the Gaussian kernel, and \( ||u - v||^2 = (u - v)'(u - v) \) is the Euclidean distance between \( u \) and \( v \). 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; (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

A.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 16kHz with 16-bit resolution. The corpus comprises a SI section and a speaker-dependent (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.

A.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, 3-state hidden Markov model (HMM) with a mixture of 10 Gaussian components per state. All Gaussians have diagonal covariances. In addition, there are a 3-state “sil” model to capture silence and a 1-state “sp” model to capture short pauses.

A.3 Experimental Procedure

Adaptation was performed using 5s and 10s of speech data (or, about 4.6s and 9.2s 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 speaker 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, 3 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 3 adaptation sets of the 12 test speakers.

For each method, three adaptation iterations were run. The system parameters, the Gaussian kernel width $\sigma$ and the regularization parameter $\beta$ 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 $\sigma$ and $\beta$ using a grid search. The set of $\sigma$ and $\beta$ 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 $\alpha$ and $\beta$. 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.

### TABLE II

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPLR</td>
<td>0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>MPLKR</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>KRR-MLLR</td>
<td>0.05</td>
<td>0.02</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>$\beta/\sigma$</th>
<th>0.001</th>
<th>0.005</th>
<th>0.01</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>64.49</td>
<td>78.60</td>
<td>83.45</td>
<td>79.58</td>
<td>74.90</td>
</tr>
<tr>
<td>0.005</td>
<td>65.26</td>
<td>77.21</td>
<td>83.40</td>
<td>81.57</td>
<td>76.73</td>
</tr>
<tr>
<td>0.01</td>
<td>64.80</td>
<td>77.28</td>
<td>83.37</td>
<td>81.93</td>
<td>77.64</td>
</tr>
<tr>
<td>0.05</td>
<td>62.28</td>
<td>78.29</td>
<td>83.06</td>
<td>83.47</td>
<td>79.89</td>
</tr>
<tr>
<td>0.1</td>
<td>66.27</td>
<td>77.88</td>
<td>83.45</td>
<td>83.78</td>
<td>80.61</td>
</tr>
<tr>
<td>0.5</td>
<td>62.98</td>
<td>77.74</td>
<td>83.42</td>
<td>83.35</td>
<td>82.58</td>
</tr>
<tr>
<td>1</td>
<td>61.97</td>
<td>77.59</td>
<td>83.69</td>
<td>83.25</td>
<td>82.82</td>
</tr>
</tbody>
</table>

### A.4 Adaptation Performance Comparison

Using the best values of $\sigma$ and $\beta$ 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 that:

- When there are only 5s of adaptation speech, only MAPLR, EMLLR, and RSW are effective; but the simple EMLLR and RSW are much better.
- When there are 10s 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.

- 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.

The better performance of MPLKR over MLLR is attributed to the exploitation of non-linearity in their framework with the use of a kernel map.

In summary, the adaptation experiments in this simple performance comparison show the effectiveness of using kernels in adaptation.
RM adaptation task show that when there are less than 5s 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 10s of adaptation data, more sophisticated adaptation methods, such as MPLKR, start to pay off although the simple RSW method performs unexpectedly well in both 5s and 10s adaptation. Table V shows the results of four common significance tests; they confirm that on the 10s 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, 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.

B.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 hours 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 8 speakers, each with about 40 utterances.

B.2 Feature Extraction and Acoustic Modeling

The traditional 39-dimensional MFCC vectors were extracted at every 10ms over a window of 25ms 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.

B.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 8 speakers (i.e., totally 320 test utterances). Each adaptation method was run for 6 iterations to study their convergence behavior. Notice that the average length of each WSJ0 utterance is 7.26s (or, 6s if one excludes the silence portions). The system parameters such as the Gaussian kernel width \( \sigma \) and regularization parameter \( \beta \) for various methods were simply adopted from the corresponding values found in RM adaptation.

B.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 10s 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 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 re-plotted in Fig. 1. It is noticed that MLLR-B does not change much with iterations, while KRR-MLLR, RSW, and MAPLR 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 3 iterations, while KRR-MLLR and MPLKR converge in about 6 iterations.

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

V. Conclusion

In this paper, we try to improve the standard maximum likelihood linear regression (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

---

**TABLE VI**

**Performance of unsupervised adaptation on WSJ. Results are word accuracies in % on the test set.**

<table>
<thead>
<tr>
<th>Model or Method</th>
<th>SI</th>
<th>MAP</th>
<th>MLLR</th>
<th>MLLR-D</th>
<th>MLLR-B</th>
<th>EMLLR</th>
<th>RSW</th>
<th>MPLLR</th>
<th>MAPLR</th>
<th>KRR-MLLR</th>
<th>MPLKR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
</tr>
<tr>
<td>Iteration</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
<td>92.60</td>
</tr>
<tr>
<td>2</td>
<td>92.75</td>
<td>92.83</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
<td>92.85</td>
</tr>
<tr>
<td>3</td>
<td>93.05</td>
<td>93.16</td>
<td>93.16</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
<td>93.20</td>
</tr>
<tr>
<td>4</td>
<td>93.16</td>
<td>93.20</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
<td>93.24</td>
</tr>
<tr>
<td>5</td>
<td>93.37</td>
<td>93.61</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
<td>93.52</td>
</tr>
<tr>
<td>6</td>
<td>93.48</td>
<td>93.56</td>
<td>93.67</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
<td>93.69</td>
</tr>
<tr>
<td>7</td>
<td>93.05</td>
<td>92.96</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
<td>92.94</td>
</tr>
<tr>
<td>8</td>
<td>92.87</td>
<td>92.98</td>
<td>93.12</td>
<td>93.23</td>
<td>93.28</td>
<td>93.28</td>
<td>93.28</td>
<td>93.28</td>
<td>93.28</td>
<td>93.28</td>
<td>93.28</td>
</tr>
<tr>
<td>9</td>
<td>93.23</td>
<td>93.37</td>
<td>93.54</td>
<td>93.62</td>
<td>93.73</td>
<td>93.73</td>
<td>93.73</td>
<td>93.73</td>
<td>93.73</td>
<td>93.73</td>
<td>93.73</td>
</tr>
<tr>
<td>10</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
<td>93.70</td>
</tr>
</tbody>
</table>
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.)

<table>
<thead>
<tr>
<th>Method</th>
<th>MP</th>
<th>SI</th>
<th>WI</th>
<th>MN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MAP</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MLLR-B</td>
<td>√</td>
<td>√</td>
<td>same</td>
<td>√</td>
</tr>
<tr>
<td>RSW</td>
<td>√</td>
<td>same</td>
<td>same</td>
<td>√</td>
</tr>
<tr>
<td>MAPLR</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>KRR-MLLR</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>MPLLR</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

![Convergence behavior of various unsupervised adaptation methods on WSJ](image)

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

as KEV, eKEV, and KEMLLR) we proposed, the new method, which we call maximum penalized likelihood kernel regression (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 about 10s 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 10s 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 10s 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.

VI. Acknowledgments

This research is partially supported by the Research Grants Council of the Hong Kong SAR under the grant numbers 617406, 617507, and 614508.

References