A Study of Minimum Classification Error (MCE) Linear Regression for Supervised Adaptation of MCE-Trained Continuous-Density Hidden Markov Models

Jian Wu, *Member, IEEE*, and Qiang Huo, *Member, IEEE*

***Abstract*—In this paper, we present a formulation of minimum classification error linear regression (MCELR) for the adaptation of Gaussian mixture continuous-density hidden Markov model (CDHMM) parameters. Two optimization approaches, namely generalized probabilistic descent (GPD) and Quickprop are studied and compared for the optimization of the MCELR objective function. The effectiveness of the proposed MCELR technique is confirmed via a series of supervised speaker adaptation experiments on a task of continuous Putonghua (Mandarin Chinese) speech recognition.**

***Index Terms*—Hidden Markov model (HMM), HMM adaptation, minimum classification error linear regression (MCELR), speaker adaptation.**

I. Introduction

SO FAR, three design principles have been widely used to construct a modern automatic speech recognition (ASR) system (readers are referred to [18] for a discussion on theoretical foundation of modern ASR formulations from a decision theoretic viewpoint):

1) using *plug-in MAP* (maximum *a posteriori* probability) as a decision rule for the recognition decision and *maximum likelihood* (ML) as a criterion for the estimation of decision parameters (e.g., [3]);
2) using *maximum discriminant* as a decision rule for the recognition decision and *minimum classification error* (MCE) as a criterion for the estimation of decision parameters (e.g., [1], [19], and [29]);
3) using *plug-in MAP* or *maximum discriminant* as a decision rule for the recognition decision and *maximum mutual information* (MMI) (e.g., [4]) or *minimum phone error* (MPE) (e.g., [36]) or *minimum word error* (MWE) (e.g., [16], [36]) as a criterion for the estimation of decision parameters.

It has been demonstrated by many research groups that when a sufficient amount of *representative* training data are available, an ASR system constructed under the second or third principle can outperform its counterpart constructed under the first principle for many ASR applications (see, for example, [7], [22], [30], [33], and [36] and the references therein). Consistent with the first design principle, many successful adaptation techniques have been developed in the past two decades to cope with the possible problem of mismatches between training and testing conditions (see, for example, a review [25] and the references therein). Consistent with the second or third design principle, MCE adaptation of parameters of Gaussian mixture continuous-density hidden Markov model (CDHMM) was first reported by authors of [28]. Several follow-up studies were also reported by other research groups (e.g., [23], [31], and [38]). One reported in [24] demonstrated that direct MCE adaptation for MCE-trained HMM parameters works well when a sufficient (*w.r.t.* the number of parameters being adapted) amount of adaptation data are available. However, when only a small amount of adaptation data are available, direct MCE adaptation of HMM parameters does not work so well. Even more recently, techniques for direct adaptation of HMM parameters under the criterion of MMI or MPE [35] were developed. Experimental results were reported on how they work for the adaptation of the ML-, MMI-, and MPE-trained seed models [35].

Inspired by the success of maximum-likelihood linear regression (MLLR) approach (e.g., [26]) for efficient HMM adaptation when only a small amount of adaptation data are available, in the past several years, there were some efforts to develop discriminative linear regression (DLR) adaptation techniques under different criteria and notions such as MCE [5], maximum scaled likelihood [42], maximal rank likelihood [10], MMI [40], and conditional maximum likelihood (CML) [13].[1] Although all of them were developed with the aim of an efficient discriminative adaptation, they were applied to adapting the ML-trained seed models in the first attempt. In our opinion, applying DLR approaches to adapting the discriminatively trained seed models is more desirable. In this way, because a consistent criterion can be used in both seed model training and the succeeding adaptation, better performance may be expected. It was this conjecture that motivated our study on MCE linear regression (MCELR) as reported in [45] and [46]. Since then, follow-up studies on MCELR were also reported (e.g., [14], [15]), where

Manuscript received May 16, 2004; revised June 11, 2006. This work was supported by the RGC of the Hong Kong SAR under Grants HKU7022/00E and HKU7039/02E. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mari Ostendorf.

J. Wu was with the Department of Computer Science, The University of Hong Kong, Hong Kong, China. He is now with the Microsoft Corporation, Redmond, WA 98052-6399 USA (e-mail: jianwu@microsoft.com).

Q. Huo is with the Department of Computer Science, The University of Hong Kong, Hong Kong, China (e-mail: qhuo@cs.hku.hk).

Digital Object Identifier 10.1109/TASL.2006.881692

[1]An extension of the work was reported in [39], where linear regression transformations were trained by using CML criterion and were applied to either feature normalization or discriminative speaker adaptive training.

different optimization approaches are used for optimizing the MCE criterion. Another recent DLR approach was reported in [6], where the LR parameters were estimated under the criterion of a so-called aggregate *a posteriori* probability originally proposed in [27] for discriminative training of Gaussian mixture models. In these studies, DLR was reported to achieve better performance than MLLR, though the DLR technique was only applied to adapting the ML-trained seed models. An MPE-based DLR approach was proposed in [43], where experimental results were reported for the adaptation of both ML- and MPE-trained seed models. It was demonstrated that the MPE-LR approach achieved better performance than MLLR in both of the experimental setups. All of the above findings are consistent with our findings in MCELR reported originally in [45], [46], and further here.

In this paper, we expand on our previous work [45] and [46], providing a more detailed description of the MCELR formulation and the algorithm derivation, additional implementation details, new experiments, and an expanded discussion of the results. The rest of the paper is organized as follows. In Section II, we extend the MCELR formulation in [5] to a more general one in the context of CDHMM and large-vocabulary ASR. In Section III, we present an alternative optimization algorithm, namely *Quickprop*, for MCELR. In Section IV, we report experimental results to demonstrate the behavior of the MCELR in a supervised speaker adaptation application. We also compare the performance achieved by the MCELR with that of MLLR. Finally, we summarize our findings in Section V.

II. MCELR Formulation

In most of linear regression-based adaptation approaches, such as MLLR (e.g., [9] and [26]), all of the Gaussian components from all CDHMMs will be clustered into several regression classes as determined by a regression class tree. Two linear transformations, $\hat{W}_m$ and $\hat{H}_m$, are assigned to a particular regression class m, which consists of R similar Gaussian components $\{m_r\}_{r=1}^{R}$. They can be used to estimate the new mean vector $\hat{\mu}_{m_r}$ and the covariance matrix $\hat{\Sigma}_{m_r}$ for the m_rth Gaussian component as follows:

$$\hat{\mu}_{m_r} = \hat{W}_m \xi_{m_r} \tag{1}$$
$$\hat{\Sigma}_{m_r} = B_{m_r}^{Tr} \hat{H}_m B_{m_r} \tag{2}$$

where $\xi_{m_r} = (1, \mu_1, \mu_2, \ldots, \mu_D)^{Tr}$ is the extended vector of a D-dimensional mean vector μ_{m_r} before transformation, B_{m_r} is the inverse of the Choleski factor of the inverse covariance matrix $\Sigma_{m_r}^{-1}$ before transformation such that

$$\begin{aligned} B_{m_r} &= C_{m_r}^{-1} \\ \Sigma_{m_r}^{-1} &= C_{m_r} C_{m_r}^{Tr}. \end{aligned} \tag{3}$$

In MLLR, an ML criterion is used for estimating the linear transformations, $\{\hat{W}_m\}$ [26] and $\{\hat{H}_m\}$ [9], from adaptation data. Of course, they can also be estimated by using other criteria, for example MCE.

In the MCE framework formulated in [20], which is the basis for our MCELR approach, a sigmoid function is adopted to approximate the empirical classification error of each training sample Y with a word string label Z_c under the current model parameter set Λ as follows:

$$l(Y; \Lambda) = \frac{1}{1 + exp\left(-\alpha d(Y; \Lambda) + \beta\right)}. \tag{4}$$

In the above equation, $d(Y; \Lambda)$ is a misclassification measure defined as follows:

$$d(Y; \Lambda) = -g(Y; \Lambda) + \bar{g}(Y; \Lambda) \tag{5}$$

where $g(Y; \Lambda)$ is a *discriminant function* for recognition decision-making, and $\bar{g}(Y; \Lambda)$ is a term referred to as *antidiscriminant function* in this paper for the convenience of reference. For a particular training sample Y, the definition of $g(Y; \Lambda)$ and $\bar{g}(Y; \Lambda)$ should satisfy that: $g(Y; \Lambda) < \bar{g}(Y; \Lambda)$ implies false classification, and $g(Y; \Lambda) > \bar{g}(Y; \Lambda)$ means correct classification. The specific forms of $g(Y; \Lambda)$ and $\bar{g}(Y; \Lambda)$ used in our MCELR formulation will be explained in detail in the next two subsections, respectively. Given the above definitions, an MCE objective function can then be formed as the empirical average loss on the training set $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_K\}$

$$\ell(\Lambda) = \frac{1}{K} \sum_{k=1}^{K} l(Y_k, \Lambda). \tag{6}$$

In [5], MCE has been used to estimate a single global linear transformation for mean parameters in a trended HMM. In the following, we present a more general version of the MCELR for estimating multiple linear transformations in a regression tree that can be used in turn for the adaptation of both mean vectors and covariance matrices of CDHMMs. Our MCELR formulation can be treated as an MCE counterpart of the traditional MLLR formulation described in [9] and [26].

In order to minimize the above MCE objective function, we use the following sequential gradient descent algorithm, coined as generalized probabilistic descent or shortly GPD by the authors of [22] due to the fact of that GPD was a slightly modified version of the probabilistic-descent (PD) method proposed originally by Amari in [1]

$$\Lambda_{k+1} = \Lambda_k - \epsilon_k \nabla l(Y_k; \Lambda)|_{\Lambda = \Lambda_k} \tag{7}$$

where k refers to the cumulative number of training samples presented so far. Let us use θ^m to denote generically the linear transformation $\hat{W}_m$ or $\hat{H}_m$. The updating formula of (7) becomes

$$\begin{aligned} \theta_{k+1}^m &= \theta_k^m - \epsilon_k \frac{\partial l}{\partial \theta^m}\bigg|_{\theta^m = \theta_k^m} \\ &= \theta_k^m - \epsilon_k \alpha l(1 - l) \left(-\frac{\partial g}{\partial \theta^m} + \frac{\partial \bar{g}}{\partial \theta^m} \right)\bigg|_{\theta^m = \theta_k^m}. \end{aligned} \tag{8}$$

Other optimization algorithms can also be used. In Section III, we describe how to use an optimization algorithm called *Quickprop* [8] for MCELR. Because most optimization algorithms need to calculate the derivatives of the discriminant and antidiscriminant functions with respect to the parameters to be optimized, in the following, we present the specific formulation of discriminant and antidiscriminant functions we used for MCELR and how to calculate their derivatives.

A. *Discriminant Function and Its Derivatives*

In our MCELR formulation, the discriminant function of a given observation Y is defined as follows:

$$g(Y;\Lambda) = \log\left(p(Y|Z_c)\right) = \log \sum_S p(Y,S|Z_c) \tag{9}$$

where S represents a possible state sequence for the given word string Z_c. To make good use of the limited adaptation data, the discriminant function in (9) considers all of possible state alignments, which is different from the one used in a conventional segmental MCE training formulation, but its rationale has been discussed in, e.g., [20]. Given an adaptation sample with a sequence of T feature vectors, $Y = (y_1, y_2, \ldots, y_T)$, the derivative of the discriminant function g with respect to θ^m is

$$\frac{\partial g}{\partial \theta^m} = \sum_{t=1}^{T}\sum_{r=1}^{R} L_{m_r}(t)\cdot \Phi(y_t, m_r) \tag{10}$$

where $L_{m_r}(t) = p(q_{m_r}(t)|Z_c, Y)$ is the occupation probability of the Gaussian component m_r at time t, and $\Phi(\cdot,\cdot)$ is a matrix that takes different forms for the following two cases.

1) If θ^m represents the mean transformation $\hat{W}_m$, then

$$\Phi(y_t, m_r) = \hat{\Sigma}_{m_r}^{-1}\left(y_t - \hat{\mu}_{m_r}\right)\cdot \xi_{m_r}^{Tr}. \tag{11}$$

In particular, if a diagonal covariance matrix is used for each Gaussian component, the above result can be further simplified as a matrix with its (u,v)th element ϕ_{uv} being

$$\phi_{uv}(y_t, m_r) = \frac{(y_{t,u} - \hat{\mu}_{m_r,u})\,\xi_{m_r,v}}{\hat{\sigma}^2_{m_r,u}} \tag{12}$$

where $\hat{\mu}_{m_r,u}$ is the uth element of the mean vector $\hat{\mu}_{m_r}$, $\xi_{m_r,v}$ is the vth element of the extended vector ξ_{m_r}, and $\hat{\sigma}^2_{m_r,u}$ is the uth diagonal element of $\hat{\Sigma}_{m_r}$.

2) If θ^m represents the covariance transformation $\hat{H}_m$, then

$$\Phi(y_t, m_r) = \frac{1}{2}\left[\hat{G}\hat{G}^{Tr} - \hat{H}_m^{-1}\right] \tag{13}$$

with $\hat{G} = \hat{H}_m^{-1}(B_{m_r}^{Tr})^{-1}(y_t - \hat{\mu}_{m_r})$.

A sketch of derivation of the above derivatives is given in Appendix I.

B. *Lattice-Based Antidiscriminant Function and Its Derivatives*

The calculation of $\partial\bar{g}/\partial\theta^m$ in (8) relates to the definition of the antidiscriminant function $\bar{g}(Y;\Lambda)$. In this subsection, we discuss how to deal with this important issue. In MCELR, there are several possible ways to define the antidiscriminant function, depending on how the competing hypotheses for a given adaptation utterance Y are chosen and used among the following three possibilities, namely, 1) the most confusable word sequence, 2) a list of N-best word sequences, and 3) a word lattice. Given the fact that only a small amount of adaptation data are available, yet the word lattice contains the richest information about the competing hypotheses with respect to the correct word sequence, we adopt a lattice-based approach to define the antidiscriminant function.

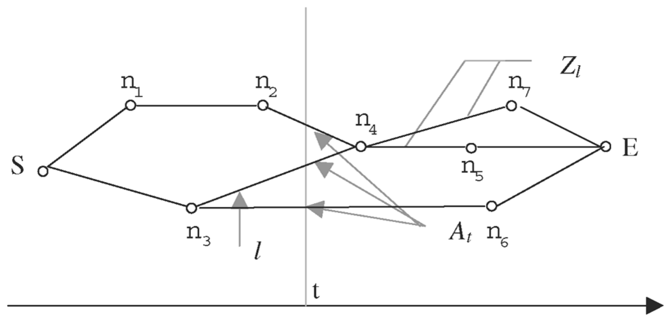

Fig. 1. Illustrative example of a word lattice for the definition of antidiscriminant function.

For each adaptation utterance Y, a "competing word lattice" as illustrated in Fig. 1 is generated by performing a recognition of Y, in which each arc l, e.g., the arc $\overrightarrow{n_3n_4}$, represents a word $z^{(l)}$ with its word boundary being $(a(l) = t_3, b(l) = t_4)$. In the simplest case, if the word boundaries are fixed during the parameter updating, the set of active arcs at arbitrary time t, denoted as A_t, can be determined in advance. In the case of the lattice illustrated in Fig. 1, $A_t = \{\overrightarrow{n_2n_4}, \overrightarrow{n_3n_4}, \overrightarrow{n_3n_6}\}$. Based on the above notations and according to the general formulation of MCE training, the lattice-based antidiscriminant function $\bar{g}(Y;\Lambda)$ in our MCELR approach can be defined, by expanding the "competing word lattice" into a long N-best list, as follows:

$$\bar{g} = \frac{1}{\eta}\log\left[\frac{\sum_Z \mathbf{1}(Z \neq Z_c)\exp\left(\eta \log p(Y|Z)\right)}{\sum_Z \mathbf{1}(Z\neq Z_c)}\right] \tag{14}$$

where

$$\mathbf{1}(Z\neq Z_c) = \begin{cases} 1, & \text{if the word sequence } Z \text{ is different} \\ & \text{from the word sequence } Z_c \\ 0, & \text{otherwise} \end{cases}$$

the summation is taken over all the possible word sequences represented by the word lattice, and η is a scaling factor. The purpose of adding this scaling factor is to adjust the contribution of different paths to the antidiscriminant function. When $\eta \to \infty$, (14) is equivalent to

$$\bar{g} = \max_{Z:Z\neq Z_c} \log p(Y|Z). \tag{15}$$

However, a straightforward implementation, using (14) to calculate the derivatives, will incur too many redundant computations simply due to the sharing of the same arc in different word sequences. An efficient algorithm making good use of the compact structure of the lattice is desirable.

It is noted that the lattice-based approach has been used in different ways in several research groups for MMI training (e.g., [37], [41], and [44]). As discussed in, e.g., [37], the situation for MCE training is slightly different from that of MMI training because the "competing word lattice" should exclude the correct word sequence. Because each arc in the lattice could be part of several word sequences, in general, a word sequence could not be excluded from the lattice without affecting other word sequences.[2]

In our MCELR approach, we define the following auxiliary probability term $P(l \notin Z_c)$ to exclude the correct word sequence from the lattice

$$P(l \notin Z_c) = \frac{\sum_{Z_l : Z_l \neq Z_c} p^\eta(Y|Z_l)}{\sum_{Z_l} p^\eta(Y|Z_l)} \tag{16}$$

where Z_l refers to a possible word sequence in the lattice passing through the arc l, for example, $\overrightarrow{Sn_3n_4n_5E}$ or $\overrightarrow{Sn_3n_4n_7E}$ in Fig. 1. Using the above notion of $P(l \notin Z_c)$, the derivative of $\bar{g}$ with respect to θ^m can be derived as sketched in Appendix II and is listed as follows:

$$\frac{\partial \bar{g}}{\partial \theta^m} = \sum_{t=1}^{T} \sum_{l \in A_t} \sum_{r=1}^{R} \bar{\varphi}(l) \bar{L}_{m_r}^{(l)}(t) \cdot \Phi(y_t, m_r) \tag{17}$$

where

$$\bar{\varphi}(l) = \frac{P(l \notin Z_c) exp(\bar{g}_l)}{\sum_{l' \in A_t} P(l' \notin Z_c) exp(\bar{g}_{l'})}. \tag{18}$$

In the above equations, $\bar{L}_{m_r}^{(l)}(t) = p(q_{m_r}(t)|z^{(l)}, Y^{(l)} = (y_{a(l)}, \cdots, y_{b(l)}))$ is the occupation probability of the Gaussian component m_r at time t with respect to the arc l; $\bar{g}_l = \log(\sum_{Z_l} p^\eta(Y|Z_l))$ is the log of the scaled total likelihood of all the word sequences passing through the arc l in the lattice, and $\Phi(.,.)$ is defined as in one of the equations in (11)–(13). Both $P(l \notin Z_c)$ and $\bar{g}_l$ can be decomposed into the forward/backward parts and be calculated by a recursive procedure similar to the ones described in, e.g., [34], [37], and [41]. In this way, the redundant computations incurred by expanding a word lattice into a long N-best list as in (14) are avoided, yet the convergence property of our MCELR approach can be justified in the same way as the conventional N-best-list based MCE training (e.g., [20]).

The rationale of using $P(l \notin Z_c)$ can be further clarified by using the following examples. Based on the above definition, if Z_c is not in the "competing word lattice," (i.e., any word sequence from S to E is different from Z_c), $P(l \notin Z_c) = 1$ for any arc l. On the other hand, if the arc l appears only in the correct word sequence, for example, $Z_c = \overrightarrow{Sn_3n_6E}$ and l denotes $\overrightarrow{n_3n_6}$, we have $P(l \notin Z_c) = 0$. In all other cases, $P(l \notin Z_c)$ takes a value between 0 and 1. Therefore, $P(l \notin Z_c)$ can be used to characterize the actual contribution of a particular arc made to the antidiscriminant function in MCE training by excluding the contribution of all correct word sequences (with different word boundaries) passing through this arc in the lattice.

With the above formulas for relevant derivatives, the updating formula in (8) can be used to update the transformation parameters. In the following subsection, we discuss several implementation issues. A good treatment of them can help the MCELR work better.

[2]A lattice-based procedure for the MCE training of CDHMMs was described briefly in [37], and experimental results were reported more recently in [30]. Inspired by the discussion on the possibility of using a lattice-based approach for MCE training in [37], we developed our own lattice-based method for MCELR as described in the following. Due to the lack of knowledge about the technical details of the lattice-based approach reported in [30] and [37], we are not able to discuss the specific differences between our treatment and the one in [30] and [37].

C. Implementation Issues

1) Initialization of Transformations: It is well known that in any gradient descent approach, a good initialization of the parameters is important to make the algorithm converge to a good local optimum. We have studied two strategies for parameter initialization. In the first approach, an identity matrix is used to initialize the relevant transformations. Given the available adaptation data, information about the occupation counts of each Gaussian component is first collected before the MCELR updating. Using this information, the regression classes can then be determined. The identity matrices are assigned to those nodes with sufficient adaptation data in the regression tree as initial values for the relevant linear transformations.

In the second approach, a supervised MLLR adaptation is performed first. Then, the regression classes suitable for MCELR are determined based on the same regression tree built by MLLR but different cutting thresholds are allowed. The initial transformation for each regression class is set as the associated MLLR transformation if it is available; otherwise, the valid MLLR transformation associated with its nearest ancestor in the regression tree is used. In this way, the MCELR can start from a more informative transformation for each regression class.

In MLLR implementation [26], there is an operation of matrix inversion in solving linear equations for estimating individual regression matrix. To avoid the possible numerical problem associated with the matrix inversion, each regression class should be "assigned" enough adaptation data. Consequently, an appropriate threshold has to be set for determining the number of regression classes that can be used for a given amount of adaptation data. However, in MCELR updating, no matrix inversion is involved. Therefore, we can use a looser threshold to determine the number of regression classes. Consequently, more transformations could be used in MCELR than that in MLLR for the same amount of adaptation data.

There are other possible ways for parameter initialization. Readers are referred to [6] for another example.

2) Scaling of Variables: For an unconstrained optimization problem, the accuracy of the solution can be greatly affected by the conditioning of the Hessian matrix at that solution (e.g., [11]). In MCELR, although it is difficult to calculate the Hessian matrix of the objective function, it is still helpful to use a diagonal scaling based on the diagonal elements of an approximate Hessian matrix. Therefore, the following scaling can be performed on the (u, v)th element, w_{uv}^m of the transformation matrix $\hat{W}_m$. For the extreme case where each mean vector has its own transformation, we can use

$$\tilde{w}_{uv}^{m_r} = w_{uv}^{m_r} \frac{\xi_{m_r,v}}{\sigma_{m_r,u}}. \tag{19}$$

As for the general case where a transformation is shared by many Gaussian components, the following approximate updating formula is used as an alternative to (12)

$$\phi_{uv}(y_t, m_r) = (y_{t,u} - \hat{\mu}_{m_r,u}) / \xi_{m_r,v}. \tag{20}$$

Another appropriate approach of scaling the transformation matrix elements is to first normalize the mean vector during the transformation as follows:

$$\tilde{\mu}_{m_r} = C_{m_r}\mu_{m_r} \tag{21}$$

where C_{m_r} is the Choleski factor of inverse covariance matrix $\Sigma_{m_r}^{-1}$ as shown in (3). Then the linear transformation $\hat{W}_m$ is applied to the normalized mean vector. The new mean vector $\hat{\mu}_{m_r}$ will be

$$\hat{\mu}_{m_r} = C_{m_r}^{-1}\hat{W}_m\tilde{\xi}_{m_r} \tag{22}$$

where $\tilde{\xi}_{m_r} = (1, \tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_D)^{Tr}$ is the extended vector of $\tilde{\mu}_{m_r}$, and $\hat{W}_m = \{w_{uv}^m\}$ is the $D \times (D+1)$ matrix to be estimated. Based on the above notations, the gradient direction $\dot{l}^m$ used for the optimization of $\hat{W}_m$, given an adaptation utterance with a sequence of T feature vectors, $Y = \{y_t\}_{t=1}^T$, is derived as

$$\begin{aligned}\dot{l}^m &\triangleq \frac{\partial l}{\partial \hat{W}_m}\\ &= \alpha l(1-l) \times \sum_{t=1}^{T}\sum_{r=1}^{R}\left[\sum_{l\in A_t}\bar{\varphi}(l)\bar{L}_{m_r}^{(l)}(t) - L_{m_r}(t)\right]\\ &\quad \cdot \Phi(y_t, m_r)\end{aligned} \tag{23}$$

where

$$\Phi(y_t, m_r) = C_{m_r}(y_t - \hat{\mu}_{m_r}) \cdot \tilde{\xi}_{m_r}^T. \tag{24}$$

Although different scaling schemes will result in different optimal values for adapted model parameters, it was observed in our speaker adaptation experiments that the performance difference between the above two scaling approaches is very small. Therefore, only the experimental results using the first scaling technique are reported in this paper.

3) Learning Rate Schedule: The learning rate ϵ_k is another important control parameter in MCELR. In this study, the following schedule is used:

$$\epsilon_{k+1} = \epsilon_k - \frac{\epsilon_0 \cdot T_k}{\sum_{k=1}^{E\cdot K} T_k} \tag{25}$$

where T_k is the number of frames in the kth cumulative training sentence, E is the total number of training epochs to be performed in MCELR (one pass of training samples is called an epoch), and ϵ_0 is a control parameter need to be carefully determined by experiments.

III. Alternative Optimization Methods for MCELR

A. First-Order Versus Second-Order Optimization

Although the efficacy of the sequential procedure like GPD for MCE training has been demonstrated by many experiments, it is not so easy to use in the optimization of MCELR. First, the sequential procedure has to be used if the training samples are not available before learning starts. However, if all samples are available in the case of batch-mode supervised adaptation, collecting the total gradient information before deciding the next step can be useful to avoid a mutual interference of the parameter changes, especially for the large-scale problems. Undoubtedly, one of the reasons in favor of the sequential approach is that it can introduce some randomness that might be helpful in escaping from a "bad" local optimum. However, there is also the risk of missing a good local optimum for the same reason. Actually, the choice of using a sequential or a batch mode optimization depends on the nature of the task under investigation. Second, GPD needs a careful tuning, via "trial-and-error," of the learning rate, to achieve a good learning behavior. Unfortunately, according to our speaker adaptation experiments on MCELR, the GPD optimization process is sensitive to the initial value ϵ_0 of the learning schedule, yet the most suitable value varies greatly for different speakers. Therefore, it is difficult to exploit the full potential of MCELR if only a fixed learning rate schedule can be used for different speakers with different amounts of adaptation data, that is unfortunately the case in speaker adaptation application.

An alternative approach for mitigating the above drawbacks of a simple gradient scheme is to use second-order information to decide the parameter updates. Usually, the second-order methods are implemented in a batch mode. For example, the batch mode Newton's method will adopt the following formula to update the parameters:

$$\omega_m(p+1) = \omega_m(p) - [H_m(p)]^{-1}\,\dot{l}_m(p) \tag{26}$$

if the $J \times J$ Hessian matrix $H_m(p)(\triangleq \nabla\dot{l}_m(p))$ is positive definite. To facilitate a clear presentation of the algorithms in this section, $\hat{W}^m$ and $\dot{l}^m$ are rewritten as the J-dimensional vectors $\omega_m = \{\omega_j^m\}_{j=1}^J$ and $\dot{l}_m = \{\dot{l}_j^m\}_{j=1}^J$ respectively, where there is a unique index pair (u, v) associated with each index j, and $J = D^2 + D$. In (26), $\dot{l}_j^m(p) = \sum_{k=1}^K \dot{l}_j^m(p,k)/K$ reflects the total gradient information of the pth epoch, where $\dot{l}_j^m(p,k)$ denotes $\dot{l}_j^m$ calculated by using the current transformed HMM parameters and the kth adaptation sample in the pth epoch. The exact implementation of such second-order methods demands a Hessian matrix that can be either calculated analytically or approximated via some computationally intensive procedures [2]. However, for an objective function like (6), the computation of the Hessian matrix is very complicated. Therefore, for large-scale optimization problems like in MCELR, some heuristic strategies have to be adopted to make the algorithms practical. Among many approaches, the Quickprop approach [8] is demonstrated to be useful in several ASR applications (e.g., [21], [32], and [33]).

B. Quickprop Algorithm

The Quickprop algorithm approximates the jth diagonal element h_{jj} of the Hessian matrix as

$$h_{jj}(p) = \frac{\partial \dot{l}_j^m(p)}{\partial \omega_j^m} \approx \frac{\dot{l}_j^m(p) - \dot{l}_j^m(p-1)}{\omega_j^m(p) - \omega_j^m(p-1)} \tag{27}$$

and treats the other off-diagonal elements as zeros. Combining (26) and (27) gives

$$\omega_j^m(p+1) = \omega_j^m(p) - \frac{\dot{l}_j^m(p)}{\dot{l}_j^m(p) - \dot{l}_j^m(p-1)}\left[\omega_j^m(p) - \omega_j^m(p-1)\right]. \tag{28}$$

The rationale behind the diagonal approximation of the Hessian matrix by the Quickprop algorithm can be explained in another way based on two "risky" assumptions made by Fahlman [8].

1) The curve of objective function with respect to each individual linear transformation parameter can be approximated by a parabola whose arms open upward.
2) The change in the slope of the curve, as seen by each individual transformation parameter, is not affected by all the other transformation parameters that are indeed changing at the same time.

Therefore, for each parameter, we can use the gradient vectors measured in the previous and current iterations, say, $\dot{l}_j^m(p-1)$ and $\dot{l}_j^m(p)$, and the related parameter update, $\Delta\omega_j^m(p-1) \triangleq \omega_j^m(p) - \omega_j^m(p-1)$, to predict independently the next step according to the upward parabola determined by these measurements. It thus needs only simple computation to jump directly to the minimum point of the suppositional parabola:

$$\begin{aligned}\Delta\omega_j^m(p) &= \kappa(p)\Delta\omega_j^m(p-1)\\ &= \frac{\dot{l}_j^m(p)}{\dot{l}_j^m(p-1) - \dot{l}_j^m(p)}\Delta\omega_j^m(p-1)\end{aligned} \tag{29}$$

which is in concordance with that of (28).

It is noted that (26) can lead to a local optimum only when the Hessian matrix H_m is positive definite and the initial point is sufficiently close to the optimum. The Quickprop also suffers from such limitations. Hence, for the case where the current gradient has the same sign as the previous gradient but has the same or even a larger magnitude, which leads to a zero or negative h_{jj}, the assumptions of Quickprop will not be true any more. If (29) is followed blindly, an unreasonable infinite or actual backward step will occur, which would end up the iteration at a point out of the trustworthy region or even lead to a local maximum. In this case, a limit on the next step size has to be set as the $\boldsymbol{\rho}$ times the previous update step. This parameter $\boldsymbol{\rho}$ is called "maximum growth factor" and usually is set as 1.75 [8]. On the other hand, when the current gradient direction $\dot{l}_j^m(p)$ is the same in sign as the previous gradient $\dot{l}_j^m(p-1)$ but is less in magnitude, $\kappa(p)$ will be positive which means the direction of the minimization has not changed yet and the previous step should be maintained in sign for the current update. When the current gradient is opposite in sign with the previous gradient, $\kappa(p)$ will be negative, which means the local optimum was crossed just now and a back step should be taken.

Another requirement in implementing Quickprop is to bootstrap the learning process when there is no previous update and gradient direction at the beginning of the first epoch. Also a similar case will occur when the process should be restarted for a particular parameter for which a step of zero was taken previously but which is now located in a nonzero-gradient region because of the change caused by the update of other parameters. In our experiments, a GPD term, where the learning rate ϵ_p tends to zero from ϵ_0 with the increase of the epochs, is used for bootstrapping purposes.

IV. Experiments and Results

A. Experimental Setup and Baseline Systems

To examine the viability and the efficacy of the proposed MCELR method, a series of experiments for continuous speech recognition of Putonghua (Mandarin Chinese) are performed. The recognition task is the recognition of 410 Putonghua base syllables disregarding tones. For the syllable language model, a uniform grammar with a syllable perplexity of 411 (i.e., each syllable can be followed by any of the 410 syllables and silence) is used. All of the recognition experiments are performed with the search engine of HTK3.0 toolkit [47]. Other details of the experimental setup are the same as that in [17] and is outlined in the following.

Two speech corpora are used for our experiments. The first speech corpus is the HKU96 Putonghua Corpus [49], which consists of a total of 20 native Putonghua speakers, ten females and ten males, each speaking hundreds of sentences from newspaper text. 18224 sentences (about 15.5 h of raw speech) from 18 speakers (nine males and nine females) are used to train two baseline speaker-independent (SI) ASR systems, one is based on ML training, and another is trained by using the MCE criterion. The second database is the 863 Putonghua Corpus acquired from mainland China [48]. Twelve speakers (six males and six females) are randomly chosen from this corpus to serve as the SI testing speakers. Among all of the 519 sentences by each speaker, 100 sentences are reserved for testing and the rest are used for adaptation.

To construct our baseline systems, for each speech frame, a 39-dimensional feature vector is generated which consists of 12 MFCCs and log-scaled energy normalized by the peak of the individual sentence, plus their first- and second-order derivatives. Sentence-based cepstral mean subtraction is applied in both training and adaptation/testing. The first baseline system is a speaker-independent decision-tree-based mixture Gaussian tied-state HMM system trained by using the ML criterion from the speech data in HKU96 corpus as described in detail in [17]. The basic speech units are the triphones considering both the within-syllable and cross-syllable contextual dependencies. Each triphone is modeled by a left-to-right three-state CDHMM. There are 3000 tied-states in total, each having four Gaussian mixture components with diagonal covariance matrices. For this baseline system, an average syllable accuracy of 60.66% is achieved over 12 testing speakers on 863 corpus. Starting from the set of SI seed models in the first baseline system, a new set of models are estimated by minimizing the embedded syllable string error rate. For each Gaussian component, only the mean vector is modified while the other parameters are kept unchanged during the MCE training. The second baseline system using this set of MCE-trained seed models achieves a syllable recognition accuracy of 64.34%. In comparison with the performance achieved by the first baseline system, a relative error rate reduction of 9.4% is achieved.

For simplicity, in all of the following speaker adaptation experiments, only mean vectors of the Gaussian components are adapted by using linear transformations. In [15], positive results were observed when MCELR was applied to adapting the variance parameters as well.

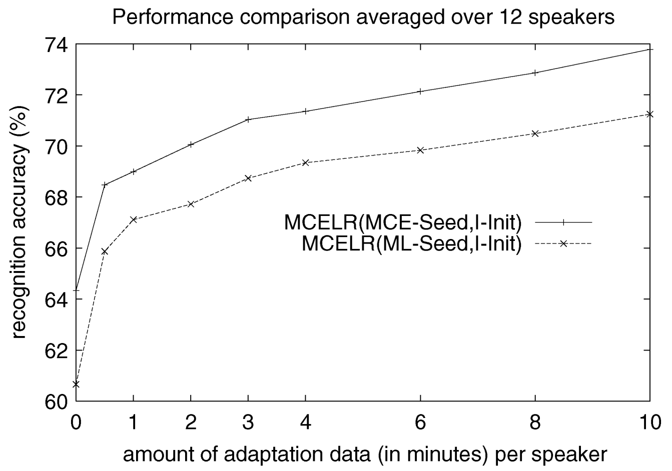

Fig. 2. Performance (syllable accuracy in percent) comparison averaged over 12 speakers with respect to the amount of adaptation data per speaker (in minutes): MCELR from ML-trained versus MCE-trained seed models.

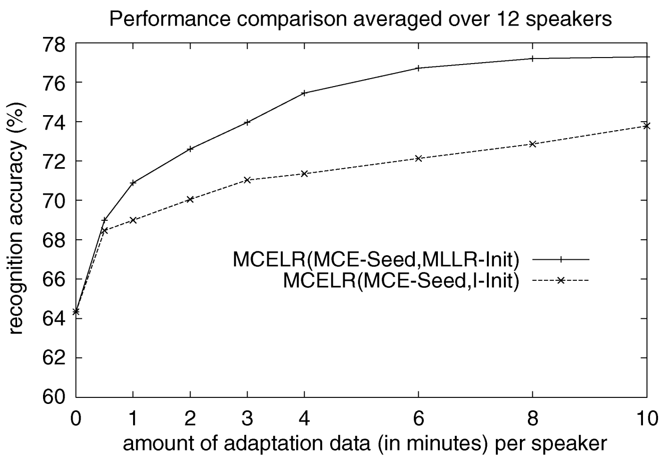

Fig. 3. Performance (syllable accuracy in percent) comparison averaged over 12 speakers with respect to the amount of adaptation data per speaker (in minutes): the effect of different initialization methods for MCELR.

B. Comparison of Performance of MLLR and MCELR Approaches for Supervised Speaker Adaptation

Starting from the above two baseline systems, we performed supervised batch-mode speaker adaptation experiments on 12 testing speakers by using both MLLR- and GPD-based MCELR. A common regression tree derived from the first baseline system and with 161 leaf nodes is used, but different numbers of transformations are chosen for MLLR and MCELR, respectively, by using different thresholds for the *occupation counts* [9], [26]. The threshold is set to be 700 for MLLR and 400 for MCELR. For all the experiments, full transformation matrices are used. The total number of training epochs are set to be 20 for each experiment of MCELR. For the experiments of MCELR starting from identity matrices, the initial learning rate ϵ_0 is set to be 0.05 while it is set to be 0.001 for MCELR starting from the transformation matrices of MLLR. The other parameters are set as $\alpha = 0.001$ and $\beta = 0$. The word lattices used for MCELR are also generated by using the HTK3.0 toolkit, which uses four tokens in each state and a uniform grammar with a syllable perplexity of 411.

In all the speaker adaptation experiments, the silence model is not adapted because there are much more silence frames than speech frames in the adaptation data we used, and a single learning schedule for MCELR is applied to all of the parameters. It was observed in some preliminary experiments that the learning process might be directed to a wrong direction if the silence model was adapted. The possible ways to address the above issue could include the following.

1) A VAD procedure is applied to adaptation sentences and only speech portions are used for model adaptation.
2) A different learning schedule is used for the silence model.
3) A specific regression class is used for the silence model only.

These could be the topics for future investigation. As a remark, the silence model was not adapted either in [6], [14], and [15].

Fig. 2 shows the performance comparison of two sets of speaker adaptation experiments on 12 speakers with respect to the amount (in minutes) of adaptation data per speaker. In the figure, "MCELR(MCE-Seed,I-Init)" and "MCELR(ML-Seed,I-Init)" refer to the MCELR adaptation starting from the MCE-trained and ML-trained CDHMM seed models, respectively, where the transformation parameters are initialized as identity matrices. It is observed that starting from both sets of SI seed models, MCELR can achieve a significant performance improvement with an increasing amount of adaptation data, yet the performance difference of two adaptation curves is maintained for different amount of adaptation data. For example, a relative error rate reduction of 8.8% is achieved by the "MCELR(MCE-Seed,I-Init)" in comparison with that of the "MCELR(ML-Seed,I-Init)" with 10-min adaptation data. By using the "simple testing approach" described in [12], the above improvement is statistically significant at the $P = 7.8 \times 10^{-7}$ level.

Fig. 3 shows the performance comparison of two sets of MCELR speaker adaptation experiments using different initialization methods for linear transformations. In the figure, both "MCELR(MCE-Seed,MLLR-Init)" and "MCELR(MCE-Seed,I-Init)" refer to the MCELR adaptation starting from the MCE-trained CDHMM seed models, in which the transformation parameters are initialized as identity matrices in the "MCELR(MCE-Seed,I-Init)" and MLLR matrices in the "MCELR(MCE-Seed,MLLR-Init)," respectively. It is observed that the "MCELR(MCE-Seed,MLLR-Init)" performs much better than the "MCELR(MCE-Seed,I-Init)." This confirms the importance of having a good initialization for the GPD-based MCELR approach.

Fig. 4 shows the performance comparison of MCELR and MLLR speaker adaptation experiments. In the figure, "MCELR(MCE-Seed,MLLR-Init)" and "MLLR(MCE-Seed,I-Init)" refer to the MCELR and MLLR adaptations, respectively, in which both approaches start from the MCE-trained CDHMM seed models, while the transformation parameters are initialized

[3]By using the testing approaches described in [12], the improvement from the "MLLR(MCE-Seed,I-Init)" to the "MCELR(MCE-Seed,MLLR-Init)" at 10 min is statistically significant at the $P = 0.071$ level according to the "simple approach," and at the $P = 0.0278$ level according to the McNEMAR's test under the assumption that half of the recognition errors made by the "MCELR(MCE-Seed,MLLR-Init)" are the same as that of the "MLLR(MCE-Seed,I-Init)."

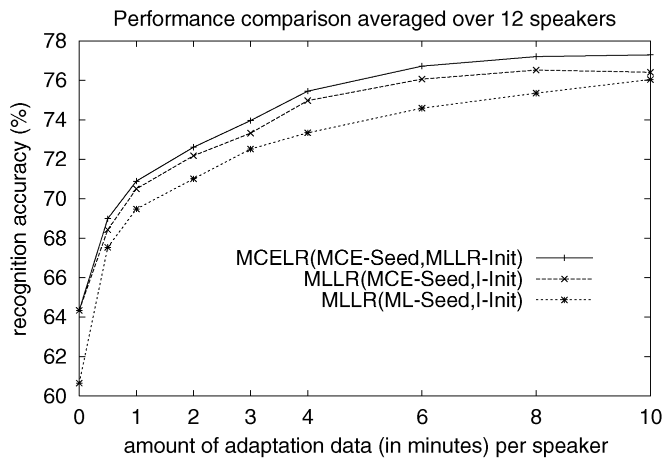

Fig. 4. Performance (syllable accuracy in percent) comparison averaged over 12 speakers with respect to the amount of adaptation data per speaker (in minutes): MLLR versus MCELR.

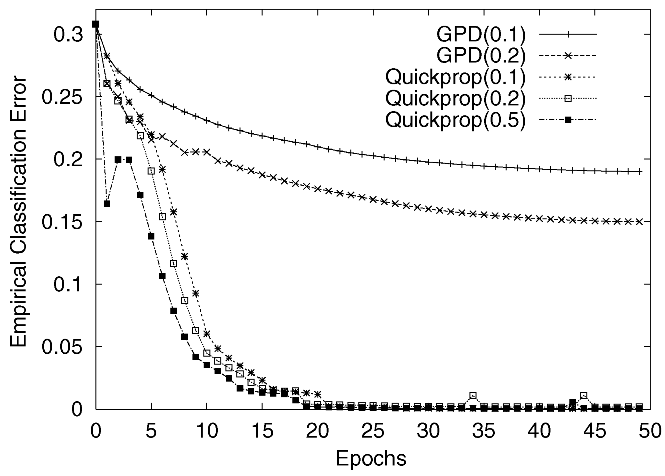

Fig. 5. Learning curves, in terms of empirical classification error, averaged over 12 speakers with respect to the number of epochs (The number within parentheses denotes the value of ϵ_0).

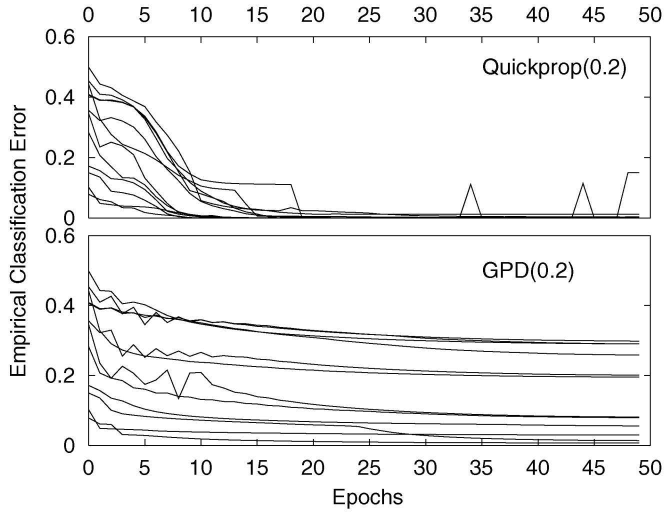

Fig. 6. Learning curves of 12 individual speakers.

as MLLR matrices in the "MCELR(MCE-Seed,MLLR-Init)" and identity matrices in the "MLLR(MCE-Seed,I-Init)," respectively. Furthermore, "MLLR(ML-Seed,I-Init)" refers to the MLLR adaptation experiments starting from ML-trained CDHMM seed models with the transformation parameters initialized as identity matrices. It is observed that, although MCE seed models perform much better than that of ML seed models, after MLLR adaptation (even with only 30-s adaptation data), the difference between two sets of adapted models (i.e. "MLLR(ML-Seed,I-Init)" versus "MLLR(MCE-Seed,I-Init)") becomes much smaller. Furthermore, by comparing the curves of "MLLR(MCE-Seed,I-Init)" and "MCELR(MCE-Seed,MLLR-Init)," it is observed that the "MCELR(MCE-Seed,MLLR-Init)" performs consistently better than the "MLLR(MCE-Seed,I-Init)." With the increasing amount of adaptation data, the gap between these two curves becomes larger, from a relative error rate reduction of 1.8% at 30s to the one of 3.7% at 10 min.[3] This is partly because more transformations can be used in MCELR than in MLLR without causing numerical problems as explained in Section II-C1. In [6], it is reported that MCELR performs better than MLLR, where both approaches start from ML-trained seed models and use the same number of regression classes.

C. Comparison of Learning Behavior of GPD and Quickprop Algorithms

Starting from the MCE-trained speaker-independent CDHMMs, we performed supervised MCELR adaptation experiments on 12 testing speakers by using both sequential-mode GPD and batch-mode Quickprop algorithms for optimization. One set of experimental results when the amount of adaptation data for each speaker is 6 min are plotted in Fig. 5. In these experiments, different values of ϵ_0 for both GPD and the bootstrapping process to start/restart Quickprop are chosen while all the other control parameters, such as α and β, are kept the same. It is observed that in terms of objective function (empirical classification error), the convergence of sequential gradient descent method is much slower than that of Quickprop algorithm, albeit the optimization of Quickprop is not fast either and sometimes oscillates during the first several iterations.

By comparing the learning behavior of GPD and Quickprop algorithms with different learning rates, it is observed that GPD is more sensitive to the ϵ_0 value than Quickprop. Furthermore, when we plot individually, in Fig. 6, the learning curves of 12 speakers, it is observed that, for Quickprop algorithm, the objective functions for all speakers are reduced to a similar low level after several epochs. However, the dynamic range of the learning curves by GPD is almost unchanged, which implies that the objective functions for some speakers may not reach their optimum, given the common parameter value ϵ_0.

D. Comparison of Performance of GPD and Quickprop Algorithms

The total number of training epochs is set to be 10 in this set of MCELR experiments. Fig. 7 shows the performance comparison of three sets of speaker adaptation experiments on 12 speakers with respect to the amount (in minutes) of adaptation data per speaker. In the figure, MCELR-Quickprop and

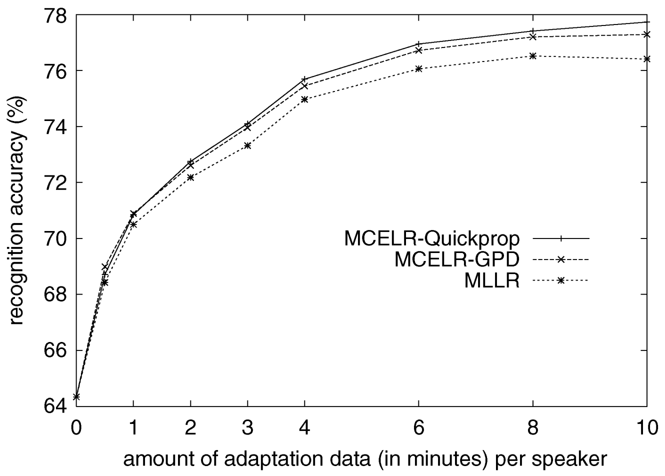

Fig. 7. Performance (syllable accuracy in percent) comparison averaged over 12 speakers with respect to the amount of adaptation data per speaker (in minutes): GPD-based versus Quickprop-based MCELR.

MCELR-GPD refer to the MCELR adaptation starting from the MCE-trained seed models by Quickprop and GPD algorithms, respectively.

It is observed that although MCELR-Quickprop decreases the objective function much more than MCELR-GPD, it cannot assure a similar impressive performance improvement in recognition accuracy. The performance of MCELR-Quickprop only outperforms slightly over that of MCELR-GPD. For the case of the small amount of adaptation data, Quickprop is even worse than GPD algorithm. It might be attributed to the "over-training" problem when the adaptation data are not sufficient and representative enough.

V. Discussion and Conclusion

We have presented a new formulation of MCELR for CDHMM adaptation. It is demonstrated that the MCELR can be used to adapt the MCE-trained HMM parameters under a consistent criterion. In a supervised speaker adaptation application, it is observed that such adapted models perform better than the ones adapted using MLLR from the ML-trained seed models. We have also presented a heuristic approach of an approximate second-order optimization algorithm, namely Quickprop, for MCELR adaptation. By a series of comparative experiments, we demonstrate that in comparison with GPD, Quickprop algorithm can achieve a faster convergence yet be less sensitive to the setting of the initial learning parameter in the optimization of the objective function for MCELR adaptation.

As a final remark, it is well known that the interaction between acoustic model and language model (LM) in discriminative training and adaptation is a very complicated issue, both theoretically and practically. We deliberately bypass this issue by not using any LM (except for syllable pronunciation) in our experiments. Although much insight has been gained on this issue via collective efforts and wisdoms of the speech community, it seems that no definite conclusion can be drawn according to experimental results reported by different research groups (e.g., [30], [33], [36], [37], and [44]). This is definitely an important research topic for future research.

Appendix I
A Sketch of Derivation of the Derivatives of Discriminant Function

Given an adaptation sample with a sequence of T feature vectors, $Y = (y_1, y_2, \ldots, y_T)$, the likelihood of Y can be expressed as

$$p(Y|\Lambda) = \sum_i \alpha_i(1)\beta_i(1) \tag{30}$$

where $\alpha_i(t) = p(y_1, \ldots, y_t, s_t = i|\Lambda)$ and $\beta_i(t) = p(y_{t+1}, \ldots, y_T|s_t = i, \Lambda)$ are the standard forward and backward terms in CDHMMs. Using the recursive update formulae of α_i and β_i, it is derived that

$$\begin{aligned}
&\sum_i \alpha_i(1)\frac{\partial \beta_i(1)}{\partial \theta^m}\\
&= \sum_i \alpha_i(1)\frac{\partial\left\{\sum_{i'} a_{ii'} b_{i'}(x_2)\beta_{i'}(2)\right\}}{\partial \theta^m}\\
&= \sum_i \sum_{i'} \alpha_i(1) a_{ii'} \beta_{i'}(2)\frac{\partial b_{i'}(y_2)}{\partial \theta^m}\\
&\quad + \sum_i \sum_{i'} \alpha_i(1) a_{ii'} b_{i'}(y_2)\frac{\partial \beta_{i'}(2)}{\partial \theta^m}\\
&= \sum_{i'} \sum_i [\alpha_i(1) a_{ii'}]\, \beta_{i'}(2)\frac{\partial b_{i'}(y_2)}{\partial \theta^m}\\
&\quad + \sum_{i'} \left\{\sum_i [\alpha_i(1) a_{ii'}]\, b_{i'}(y_2)\right\}\frac{\partial \beta_{i'}(2)}{\partial \theta^m}\\
&= \sum_{i'} \frac{\alpha_{i'}(2)\beta_{i'}(2)}{b_{i'}(y_2)}\frac{\partial b_{i'}(y_2)}{\partial \theta^m} + \sum_{i'} \alpha_{i'}(2)\frac{\partial \beta_{i'}(2)}{\partial \theta^m}\\
&= \sum_{t=2}^{T} \sum_{i'} \frac{\alpha_{i'}(t)\beta_{i'}(t)}{b_{i'}(y_t)}\frac{\partial b_{i'}(y_t)}{\partial \theta^m}\\
&\quad + \sum_{i'} \alpha_{i'}(T)\frac{\partial \beta_{i'}(T)}{\partial \theta^m}.
\end{aligned} \tag{31}$$

Also, since $\alpha_i(1) = \pi_i b_i(y_1)$ and $\beta_i(T) = a_{iN}$, we can have

$$\begin{aligned}
\frac{\partial p(Y|\Lambda)}{\partial \theta^m} &= \sum_i \left[\beta_i(1)\frac{\partial \alpha_i(1)}{\partial \theta^m} + \alpha_i(1)\frac{\partial \beta_i(1)}{\partial \theta^m}\right]\\
&= \sum_i \pi_i \beta_i(1)\frac{\partial b_i(y_1)}{\partial \theta^m}\\
&\quad + \sum_{t=2}^{T} \sum_{i'} \frac{\alpha_{i'}(t)\beta_{i'}(t)}{b_{i'}(y_t)}\frac{\partial b_{i'}(y_t)}{\partial \theta^m}\\
&\quad + \sum_{i'} \alpha_{i'}(T)\frac{\partial \beta_{i'}(T)}{\partial \theta^m}\\
&= \sum_i \left[\alpha_i(1)\beta_i(1)\frac{1}{b_i(y_1)}\frac{\partial b_i(y_1)}{\partial \theta^m}\right]\\
&\quad + \sum_{t=2}^{T} \sum_{i'} \left[\alpha_{i'}(t)\beta_{i'}(t)\frac{1}{b_{i'}(t)}\frac{\partial b_{i'}(y_t)}{\partial \theta^m}\right]\\
&= \sum_{t=1}^{T} \sum_i \alpha_i(t)\beta_i(t)\frac{1}{b_i(y_t)}\frac{\partial b_i(y_t)}{\partial \theta^m}.
\end{aligned} \tag{32}$$

Based on the general formulation of MCELR, the sate-dependent output probability density function $b_i(y_t)$ is represented by a mixture of Gaussian components

$$\begin{aligned} b_i(y_t) &= \sum_j c_{ij}\mathcal{N}(y_t; \hat{\mu}_{ij}, \hat{\Sigma}_{ij}) \\ &= \sum_j \frac{c_{ij}}{(2\pi)^{D/2}|\hat{\Sigma}_{ij}|^{1/2}} \\ &\quad \times \exp\left(-\frac{1}{2}(y_t - \hat{\mu}_{ij})^{Tr}\hat{\Sigma}_{ij}^{-1}(y_t - \hat{\mu}_{ij})\right). \end{aligned}$$

Therefore, the derivative of the discriminant function g with respect to $\hat{W}^m$ can be derived as follows:

$$\begin{aligned} \frac{\partial g}{\partial \hat{W}^m} &= \sum_{t=1}^{T}\sum_i\sum_j \left[\frac{\alpha_i(t)\beta_i(t)}{p(Y|\Lambda)}\right]\left[\frac{c_{ij}\mathcal{N}(y_t;\hat{\mu}_{ij},\hat{\Sigma}_{ij})}{b_i(y_t)}\right] \\ &\quad \times \mathbf{1}\left(j \in \{m_r\}\right)\hat{\Sigma}_{ij}^{-1}(y_t - \hat{\mu}_{ij})\xi_{ij} \\ &= \sum_{t=1}^{T}\sum_{r=1}^{R} L_{m_r}(t)\cdot\hat{\Sigma}_{ij}^{-1}\left(y_t - \hat{\mu}_{m_r}\right)\xi_{m_r}, \end{aligned} \tag{33}$$

where

$$\mathbf{1}(j \in m_r) = \begin{cases} 1, & \text{if the } j\text{th Gaussian component} \\ & \text{belongs to the regression class } m \\ 0, & \text{otherwise} \end{cases}$$

and $L_{m_r}(t) = p(q_{m_r}(t)|Z_c, Y)$ is the occupation probability of the Gaussian component m_r at time t.

The derivative of the discriminant function g with respect to $\hat{H}_m$ can be derived similarly by making use of the relevant rules in matrix calculus and the symmetric property of the covariance matrix $\hat{\Sigma}$ and the transformation matrix $\hat{H}_m$. The result is as follows:

$$\begin{aligned} \frac{\partial g}{\partial \hat{H}^m} &= -\frac{1}{2}\sum_{t=1}^{T}\sum_{r=1}^{R} L_{m_r}(t) \\ &\quad \cdot\left\{\hat{H}_m^{-1} - \left[\hat{H}_m^{-1}\left(B_{m_r}^{Tr}\right)^{-1}(y_t - \hat{\mu}_{m_r})\right]\right. \\ &\quad \left. \times\left[\hat{H}_m^{-1}\left(B_{m_r}^{Tr}\right)^{-1}(y_t - \hat{\mu}_{m_r})\right]^{Tr}\right\}. \end{aligned}$$

Appendix II
A Sketch of Derivation of the Derivatives of Lattice-Based Antidiscriminant Function

Given the definition of the lattice-based antidiscriminant function $\bar{g}$ in (14), its derivative with respect to the linear regression parameters $\theta^{(m)}$ is

$$\begin{aligned} \frac{\partial \bar{g}}{\partial \theta^m} &= \frac{1}{\eta}\sum_Z \frac{\mathbf{1}(Z \neq Z_c)\partial p^\eta(Y|Z)/\partial\theta^m}{\sum_{Z'}\mathbf{1}(Z' \neq Z_c)p^\eta(Y|Z')} \\ &= \sum_Z \frac{\mathbf{1}(Z \neq Z_c)\left[p^\eta(Y|Z)/p(Y|Z)\right]\left[\partial p(Y|Z)/\partial\theta^m\right]}{\sum_{Z'}\mathbf{1}(Z' \neq Z_c)p^\eta(Y|Z')}. \end{aligned}$$

Then, according to (32), $\partial\bar{g}/\partial\theta^m$ becomes

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\sum_Z \frac{\mathbf{1}(Z \neq Z_c)p^\eta(Y|Z)}{\sum_{Z'}\mathbf{1}(Z' \neq Z_c)p^\eta(Y|Z')}\bar{L}_{m_r}^{(Z)}(t)\Phi(y_t, m_r)$$

where $\bar{L}_{m_r}^{(Z_l)}(t) = p(q_{m_r}(t)|Z_l, Y)$ is the occupation probability of the Gaussian component m_r at time t for hypothesized word sequence Z_l, and $\Phi(\cdot,\cdot)$ represents one of the matrices as shown in (11)–(13).

It is noted that given the definition of $P(l \notin Z_c)$ in (16), the following equation is always true for an arbitrary time t:

$$\begin{aligned} \sum_Z \mathbf{1}(Z \neq Z_c)p^\eta(Y|Z) &= \sum_{l\in A_t}\sum_{Z_l: Z_l \neq Z_c} p^\eta(Y|Z_l) \\ &= \sum_{l\in A_t} P(l \notin Z_c)\exp(\bar{g}_l) \end{aligned} \tag{34}$$

where $\bar{g}_l = \log(\sum_{Z_l} p^\eta(Y|Z_l))$ is the log of the scaled total likelihood of all the word sequences passing through the arc l in the lattice. Therefore, by reordering and combining all the items in the above equations, the formula in (17) can be derived as follows:

$$\begin{aligned} \frac{\partial \bar{g}}{\partial \theta^m} &= \sum_{t=1}^{T}\sum_{r=1}^{R}\sum_{l\in A_t} \frac{P(l \notin Z_c)\exp(\bar{g}_l)}{\sum_{l'\in A_t} P(l' \notin Z_c)\exp(\bar{g}_{l'})} \\ &\quad \times \bar{L}_{m_r}^{(l)}(t)\Phi(y_t, m_r) \end{aligned}$$

where $\bar{L}_{m_r}^{(l)}(t) = p(q_{m_r}(t)|z^{(l)}, Y^{(l)} = (y_{a(l)}, \cdots, y_{b(l)}))$ is the occupation probability of the Gaussian component m_r at time t with respect to the arc l.

Acknowledgment

The authors would like to thank the reviewers and Prof. M. Ostendorf who provided constructive comments that improved the presentation of this paper.

References

[1] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 3, pp. 299–307, Jun. 1967.

[2] R. Battiti, "First- and second-order methods for learning: between steepest descent and Newton's method," *Neural Comput.*, vol. 4, pp. 141–166, 1992.

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 2, pp. 179–190, Mar. 1983.

[4] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, Tokyo, Japan, Apr. 1986, pp. 49–52.

[5] R. Chengalvarayan, "Speaker adaptation using discriminative linear regression on time-varying mean parameters in trended HMM," *IEEE Signal Process. Lett.*, vol. 5, no. 3, pp. 63–65, Mar. 1998.

[6] J.-T. Chien and C.-H. Huang, "Aggregate a posteriori linear regression adaptation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 797–807, May 2006.

[7] W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proc. IEEE*, vol. 88, no. 8, pp. 1201–1223, Aug. 2000.

[8] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-88-162, 1988.

[9] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR Framework," *Comput. Speech Lang.*, vol. 10, pp. 249–264, 1996.

[10] Y.-Q. Gao, Y.-X. Li, and M. Picheny, "Maximal rank likelihood as an optimization function for speech recognition," in *Proc. ICSLP*, Beijing, China, Oct. 2000, vol. 4, pp. 125–128.

[11] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Norwell, MA: Academic, 1981.

[12] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, 1989, pp. 532–535.

[13] A. Gunawardana and W. Byrne, "Discriminative speaker adaptation with conditional maximum likelihood linear regression," in *Proc. Eurospeech*, Aalborg, Denmark, 2001, pp. 1203–1206.

[14] X.-D. He and W. Chou, "Minimum classification error linear regression for acoustic model adaptation of continuous density HMMs," in *Proc. ICASSP*, Hong Kong, China, 2003, pp. I-556–I-559.

[15] ——, "Minimum classification error (MCE) model adaptation of continuous density HMMs," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 1629–1632.

[16] G. Heigold, W. Macherey, R. Schluter, and H. Ney, "Minimum exact word error training," in *Proc. ASRU*, 2005, pp. 186–190.

[17] Q. Huo and B. Ma, "On-line adaptive learning of continuous density hidden Markov models based on multiple-stream prior evolution and posterior pooling," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 388–398, May 2001.

[18] Q. Huo, "A decision theoretic formulation for robust automatic speech recognition," in *Pattern Recognition in Speech and Language Process.*, W. Chou and B.-H. Juang, Eds. Boca Raton: CRC, 2003, pp. 81–114.

[19] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[20] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[21] S. Kapadia, V. Valtchev, and S. J. Young, "MMI training for continuous phoneme recognition on the TIMIT database," in *Proc. ICASSP*, 1993, pp. II-491–II-494.

[22] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, Nov. 1998.

[23] F. Korkmazskiy and B.-H. Juang, "Discriminative adaptation for speaker verification," in *Proc. ICSLP*, 1996, pp. III-28–III-31.

[24] K. Laurila, M. Vasilache, and O. Viikki, "A combination of discriminative and maximum likelihood techniques for noise robust speech recognition," in *Proc. ICASSP*, 1998, pp. 85–88.

[25] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proc. IEEE*, vol. 88, no. 8, pp. 1241–1269, Aug. 2000.

[26] 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.

[27] Q. Li and B.-H. Juang, "Fast discriminative training for sequential observations with application to speaker identification," in *Proc. ICASSP*, Hong Kong, China, 2003, pp. II-217–II-220.

[28] C.-H. Lin, C.-H. Wu, and P.-C. Chang, "A study on speaker adaptation for Mandarin syllable recognition with minimum error discriminative training," *IEICE Trans. Inf. Syst.*, vol. E78-D, no. 6, pp. 712–718, 1995.

[29] A. Ljolje, Y. Ephraim, and L. R. Rabiner, "Estimation of hidden Markov model parameters by minimizing empirical error rate," in *Proc. ICASSP*, 1990, pp. 709–712.

[30] W. Macherey, L. Haferkamp, R. Schluter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Interspeech*, Lisbon, Portugal, 2005, pp. 2133–2136.

[31] T. Matsui and S. Furui, "A study of speaker adaptation based on minimum classification error training," in *Proc. Eurospeech*, Madrid, Sep. 1995, pp. 81–84.

[32] E. McDermott, "Discriminative training for speech recognition," Ph.D. dissertation, School Eng., Waseda Univ., Tokyo, Japan, 1997.

[33] E. McDermott, T. J. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, Jan. 2007.

[34] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-based unsupervised MLLR for speaker adaptation," in *Proc. ISCA ITRW ASR*, Paris, France, Sep. 2000, pp. 128–132.

[35] D. Povey, M. J. F. Gales, D. Y. Kim, and P. C. Woodland, "MMI-MAP and MPE-MAP for acoustic model adaptation," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 1981–1984.

[36] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2004.

[37] R. Schlüter, W. Macherey, B. Müller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Commun.*, vol. 34, pp. 287–310, May 2001.

[38] J. Takahashi and S. Sagayama, "Discriminative training based on minimum classification error for a small amount of data enhanced by vector-field-smoothed Bayesian learning," *IEICE Trans. Inf. Syst.*, vol. E79-D, no. 12, pp. 1700–1707, 1996.

[39] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalization and speaker adaptation in HMM estimation," in *Proc. ICSLP*, Denver, CO, 2002, pp. 2585–2588.

[40] L. Uebel and P. C. Woodland, "Improvements in linear transform based speaker adaptation," in *Proc. ICASSP*, 2001, pp. I-49–I-52.

[41] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young, "MMIE training of large vocabulary recognition systems," *Speech Commun.*, vol. 22, pp. 303–314, 1997.

[42] R. Wallhoff, D. Willett, and G. Rigoll, "Frame-discriminative and confidence-driven adaptation for LVCSR," in *Proc. ICASSP*, 2000, pp. 1835–1838.

[43] L. Wang and P. C. Woodland, "MPE-based discriminative linear transform for speaker adaptation," in *Proc. ICASSP*, 2004, pp. I-321–I-324.

[44] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, pp. 25–47, 2002.

[45] J. Wu and Q. Huo, "Supervised adaptation of MCE-trained CDHMMs using minimum classification error linear regression," in *Proc. ICASSP*, Orlando, FL, 2002, pp. I-605–I-608.

[46] ——, "A comparative study of quickprop and GPD optimization algorithms for MCELR adaptation of CDHMM parameters," in *Proc. Int. Symp. Chinese Spoken Lang. Process.*, Taipei, Tawain, R.O.C., Aug. 2002, pp. 355–358.

[47] S. Young *et al.*, The HTK Book (for HTK V3.0) Jul. 2000.

[48] Y.-Q. Zu, "Sentences design for speech synthesis and speech recognition database by phonetic rules," in *Proc. Eurospeech*, 1997, pp. 743–746.

[49] Y.-Q. Zu, W.-X. Li, M.-C. Ho, and C. Chan, "HKU96—A Putonghua Corpus" (CDROM version), Speech Lab., Dept. Comput. Sci., Univ. Hong Kong, Hong Kong, China, 1996, Tech. Memo.

Jian Wu (M'05) received the B.S. and M.S. degrees in computer science from Tsinghua University, Beijing, China, in 1998 and 2000, respectively, and the Ph.D. degree in computer science from the University of Hong Kong (HKU) in 2004.

From 1997 to 2000, he was with the Center of Speech Technology, Tsinghua University. From 2000 to 2004, he was a member of the Human–Machine Communication Laboratory, HKU. In 2004, he joined the Speech and Natural Language Group, Microsoft Corporation, Redmond, WA, working on speech recognition for desktop, telephony server, and other devices. His current research interests include speech recognition in adverse acoustical environments, acoustic modeling, front-end processing, and machine learning for speech recognition.

Qiang Huo (M'95) received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree from the USTC in 1994, all in electrical engineering.

From 1986 to 1990, his research work focused on the hardware design and development for real-time digital signal processing, image processing and computer vision, and speech and speaker recognition. From 1991 to 1994, he was with the Department of Computer Science, the University of Hong Kong (HKU), where he conducted research on speech recognition. From 1995 to 1997, he was with ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan, where he engaged in research in speech recognition. He rejoined the Department of Computer Science, HKU, in 1998 and has been an Associate Professor there since 2001. His current major research interests include automatic speech recognition; handwriting recognition; user authentication via voice, fingerprint, and their combination; Chinese/English OCR; computational model for spoken-language processing; multimodal human–machine communication; adaptive signal modeling and processing; general pattern recognition theory; machine learning; and information retrieval.