Rapid Speaker Adaptation Using Clustered Maximum Likelihood Linear Basis with Sparse Training Data

Yun Tang and Richard Rose, Senior Member, IEEE

Abstract—Speaker space based adaptation methods for automatic speech recognition have been shown to provide significant performance improvements for tasks where only a few seconds of adaptation speech is available. However, these techniques are not widely used in practical applications because they require large amounts of speaker dependent training data and large amounts of computer memory. The authors propose a robust, low complexity technique within this general class that has been shown to reduce word error rate, reduce the large storage requirements associated with speaker space approaches, and eliminate the need for large numbers of utterances per speaker in training. The technique is based on representing speakers as a linear combination of clustered linear basis vectors and a procedure is presented for maximum likelihood estimation of these vectors from training data. Significant word error rate reduction was obtained using these methods relative to speaker independent performance for the Resource Management and Wall Street Journal task domains.

Index Terms—eigenvoices, cluster adaptive training, parameter tying, speech recognition, speaker adaptation.

I. INTRODUCTION

Adaptation of hidden Markov models (HMMs) in automatic speech recognition (ASR) has been performed under a variety of scenarios. These scenarios differ in terms of the level of supervision available, the amount and type of adaptation data, and the level of computational resources available during adaptation. This paper is concerned with supervised and unsupervised adaptation scenarios where only a very small amount of adaptation speech is available. However, these techniques have little effect on performance when only a few seconds of adaptation data is available for adapting acoustic models. Speaker space based techniques have been shown to be the most successful at achieving significant reductions in ASR word error rate (WER) when using very short adaptation utterances. This is because all sources of variability are modeled within a low dimensional subspace. Techniques in this general class were originally presented for speaker adaptation under the headings of eigenvoice modeling [4]–[6], reference speaker weighting (RSW) [7]–[9], and cluster adaptive training (CAT) [10]. Many related techniques that are also based on low dimensional speaker space representations have been presented more recently [11]–[17].

Typically, speaker space based adaptation is performed on the supervector \( \vec{\mu} = (\vec{\mu}_1, \vec{\mu}_2, ..., \vec{\mu}_M)' \) formed by concatenating the mean vectors associated with all Gaussian densities in a continuous Gaussian mixture density HMM. Each of the vectors \( \vec{\mu}_m \in \mathbb{R}^D \) is a mean vector for the \( m \)th Gaussian density and it is not unusual for there to be \( M = 100,000 \) densities in a large vocabulary HMM system. This can result in an over all dimensionality, \( MD \), of over a million. The speaker space procedures estimate a set of basis vectors, \( \vec{e}(k), k = 1, \ldots, K \), from training data where \( K << MD \) and typically lies in the range \( 10 < K < 100 \). During the adaptation stage, a \( K \) dimensional weight vector, \( \vec{\omega} = (w_1, \ldots, w_K)' \), is estimated from the adaptation data and the adapted supervector, \( \hat{\vec{\mu}} \), is computed as

\[
\hat{\vec{\mu}} = \vec{\mu} + \sum_{k=1}^{K} w_k \cdot \vec{e}(k). \tag{1}
\]

The above procedure consists of two separate stages for training and adaptation. In the training stage, the set of basis vectors in a low dimension subspace is identified from training data by using one of several modeling criteria. In the adaptation stage, new HMM parameters are obtained, as shown in Equation 1, as a weighted linear combination of these basis vectors where the weights are estimated from the adaptation utterances. There are several methods that have been used to estimate the basis vectors. Kuhn et al were the first to estimate low dimensional speaker space basis vectors by first training speaker dependent (SD) HMM models and then performing principal components analysis (PCA) on...
speaker specific supervectors derived from these models [4], [18]. Basis vectors derived using this method are referred to as eigenvoices [4]. A second method for estimating the basis vectors is to use the Expectation-Maximization algorithm to maximize the likelihood of the training data with respect to the basis vectors and an initial estimate of the weight vectors in Equation 1 [5], [10]. A third method for basis vector estimation is through a probabilistic formulation for PCA using a Gaussian latent variable model known as probabilistic principal components analysis (PPCA) [19], [20]. The techniques used in the training phase of this work will be shown in Section III-C to be most closely related to the ML based approaches.

The process used for the weight vector estimation during adaptation is very similar for all of the speaker space based adaptation approaches. The EM algorithm is used to obtain ML estimates of the weight vector where it is assumed for each iteration of the EM algorithm that the basis vectors are known [4], [10]. This same procedure is used here in the adaptation phase for estimating weights in Equation 1.

Most of the speaker space methods suffer in some way from two issues. The first issue relates to problems with data sparsity in estimating the basis vectors, \( e(k) \), \( k = 1, \ldots, K \), and determining the speaker space dimension in the training stage. This problem is particularly severe in the eigenvoice approaches since they require sufficient data from each training speaker to provide statistically robust estimation of speaker specific supervectors for PCA estimation of the basis vectors. The second issue relates to the heavy storage requirements associated with these methods in the adaptation stage. The memory overhead for storage of the MD dimensional basis vectors, \( e(k) \), \( k = 1, \ldots, K \), is made particularly severe by the high dimensionality of the original feature space which can easily equal \( MD = 4 \) million parameters. Even for a \( K = 10 \) dimensional speaker space, the storage requirements for the basis vectors in Equation 1 can exceed 160 Mbytes. Chen et al proposed an alternative approach which also has the effect of reducing the storage requirements for the basis vectors [13]. In their approach, the speaker space is spanned by eigen-matrices from speaker specific MLLR transformation matrices [13]. However, the high requirement for the speaker specific training data is not solved and still an obstacle to practical applications.

This paper presents techniques that both reduce the amount of data required in the training stage of this process and also reduce the storage requirements during the adaptation stage. There are three contributions associated with the work described here. The first is a speaker subspace approach that relies on “clustered basis vectors.” The basis vectors are formed by concatenating \( N \) clustered subvectors where \( N \) is much less than \( M \), the number of HMM Gaussian densities. One effect of these clustered basis vectors is to improve statistical robustness in both training and test stages. Another practical effect is to reduce the considerable storage requirements associated with storing the basis vectors by over an order of magnitude. The second contribution is an expectation-maximization (EM) algorithm for maximum likelihood estimation of basis vectors for the unclustered and clustered cases from the training data. This representation will be referred to in the paper as a maximum likelihood linear basis (MLLB). Finally, the third contribution of the paper is an experimental study demonstrating that the success of this general class of techniques is due to its ability in practice to characterize general speech variability rather than interspeaker variability in particular. The impact of this result is very important since it obviates the need for large amounts of speaker specific training data as is generally required for other speaker space approaches.

The rest of the paper is organized as follows. A brief introduction to eigenvoice-based speaker adaptation is given in Section II. In Section III, the EM algorithm for estimating the MLLB vectors in a clustered acoustic space will be presented. Implementation issues, including the representation level of the speaker space and an analysis of convergence issues for the above EM algorithm will be discussed in Section IV. In Section V, the results of an experimental study comparing the methods introduced here with a variant of the eigenvoice modeling approach for supervised and unsupervised speaker adaptation on the Resource Management (RM) [21] task will be presented. Experimental results will also be presented for unsupervised adaptation on the Wall Street Journal (WSJ) corpus [22]. Finally, discussion and conclusions are given in Sections VI and VII.

II. EIGENVOICE SPEAKER ADAPTATION

As originally proposed by Kuhn et al, the eigenvoice method for speaker space based speaker adaptation begins by training a set of SD continuous mixture density HMMs for each of a relatively large population of speakers [4]. A set of basis vectors are computed using PCA applied to the supervectors derived from these SD HMM’s as described in Section I. Adapting the HMM to the observation vectors, \( O^s = \bar{o}^1_1 \ldots \bar{o}^T_r \), from speaker \( s \) is performed by estimating the weights, \( \bar{\omega}^s = (w^1_k, \ldots, w^K_k) \), and updating the supervector, \( \hat{\mu} \), as shown in Equation 1. A maximum likelihood estimate of the weight vector, \( \bar{\omega}^s \), is obtained by maximizing the auxiliary Q function

\[
Q(\Lambda, \hat{\Lambda}) = - \sum_{m=1}^{M} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s) (\hat{o}^s_t - \hat{\mu}^s_m)^\top \Sigma_m^{-1} (\hat{o}^s_t - \hat{\mu}^s_m) \tag{2}
\]

where

\[
\hat{\mu}^s_m = \hat{\mu}^s_m + \sum_{k=1}^{K} w^s_k \tilde{e}(k, m) \tag{3}
\]

and \( \Lambda \) and \( \hat{\Lambda} \) correspond to the initial and reestimated versions of \( \hat{\mu} \). In Equation 2, \( \lambda_m(t, s) \) is the occupation probability of the \( m \)th Gaussian density for observation vector \( \bar{o}^s_t \). \( \mu_m \) and \( \Sigma_m \) are the mean vector and covariance matrix of Gaussian \( m \), and \( \tilde{e}(k, m) \) is a subvector of basis vector \( \bar{e}(k) \) corresponding to the \( m \)th Gaussian in the HMM.

By maximizing \( Q(\Lambda, \hat{\Lambda}) \) with respect to \( \bar{\omega}^s \), it can be shown that the optimum weight vectors can be obtained by solving the following matrix equation

\[
\bar{\omega}^s = A^{-1} \hat{b} \tag{4}
\]
where
\[ b_t = \sum_{m} \sum_{t} \lambda_{m}(t, s) \vec{e}(i, m) \Sigma_{m}^{-1}(\vec{e}_t - \vec{\mu}_m) \]  
\[ a_{i,j} = \sum_{m} \sum_{t} \lambda_{m}(t, s) \vec{e}(i, m) \Sigma_{m}^{-1} \vec{e}(j, m) \]
and \( \vec{b} = (b_1, b_2, \ldots, b_K)' \) and \( A = \{a_{i,j}\}, i, j = 1, \ldots, K \). This procedure is referred to as maximum likelihood eigen-decomposition (MLED) in [4]. A major issue associated with this method concerns the large number of training utterances required from each speaker for PCA estimation of the basis vectors. Another issue is the high dimensionality of these basis vectors as discussed in Section I. These issues are addressed below in Sections III and IV.

### III. Clustered Maximum Likelihood Linear Basis

To make the processes of training and adaptation in speaker space methods more efficient and robust, one can exploit the fact that there is a great deal of redundancy in the supervector representation described in Section I. The dimensionality of the basis vectors in Equation 3 can be extremely large. The acoustic HMM trained for the large vocabulary continuous speech recognition (LVCSR) task described in Section V contains over 200,000 Gaussians, resulting in 200,000 subvector means in Equation 3 and a dimensionality of approximately eight million for the associated supervectors and basis vectors. It is natural to question whether it is necessary to work directly in such a high dimensional space and whether it might be possible to improve the statistical robustness and memory efficiency of the subspace identification methods.

This section describes an attempt to exploit this redundancy in the supervector representation by forming equivalence classes of subvector means where an equivalence class \( \phi(m) \) for subvector mean \( \vec{\mu}_m \) is obtained through a clustering procedure. A ML procedure for estimating the basis vectors, \( \vec{e}(k) \), is presented in Section III-C as a model based alternative to the PCA method for identifying a low dimensional subspace that captures relevant sources of variability.

#### A. Subvector Equivalence Classes

The speaker space methods can be described by a generative model for an observation vector, \( \vec{e}_t \), uttered by speaker \( s \). It is assumed that at each time \( t \) a mixture component \( m(t) \) for the speaker independent (SI) HMM is generated and the observation vector at time \( t \) corresponds to a speaker dependent variation about the subvector mean \( \vec{\mu}_{m(t)} \). A SI residual error term, \( \vec{e}_{m(t)} \), represents the observation error for the mixture resulting in the generative model,

\[ \vec{e}_t = \vec{\mu}_{m(t)} + \sum_{k=1}^{K} w_k \vec{e}(k, m(t)) + \vec{e}_{m(t)} \]  
(7)

where \( \vec{e}_{m(t)} \sim \mathcal{N}(0, \Sigma_{m(t)}) \).

The generative model in Equation 7 can be made more robust by associating subvectors of the basis vectors with equivalence classes of the subvector means, \( \vec{\mu}_m \), in the following steps. First, all subvectors of the supervector \( \vec{\mu} \) are clustered so that each subvector is associated with class \( \phi(m) \). Second, all subvectors \( \vec{e}(k, \phi(m)) \) of the basis vectors in Equation 7 are associated with this equivalence class rather than with the individual Gaussian. This amounts to tying the basis subvectors as \( \vec{e}(k, \phi(m)) \), so that the generative model can be written as

\[ \vec{e}_t = \vec{\mu}_{m(t)} + \sum_{k=1}^{K} w_k \vec{e}(k, \phi(m(t))) + \vec{e}_{m(t)}. \]  

(8)

The number of physical subvectors, \( N \), is much smaller than than the number of logical subvectors, \( M \). It will be shown in Section V that a mapping of physical subvector to logical subvector where \( M/N \approx 10 \) represents a reasonable trade-off between acoustic sensitivity and statistical robustness. The vectors \( \vec{e}(k, \phi(m)) \) will be referred to as a clustered maximum likelihood linear basis (CMLLB).

![Speaker adaptation using the CMLLB.](image)

#### B. Identifying Subvector Equivalence Classes

The goal in forming equivalence classes of mixture Gaussians is to cluster densities that behave similarly when subjected to sources of variability in the data. This removes a degree of redundancy in the representation of the basis vectors. It is reasonable to assume that similar Gaussian densities form homogeneous regions in the acoustic space and that sources of variability will tend to have a similar affect on all Gaussians in a given region. Therefore, clustering densities into regions
using a likelihood based similarity measure will also have the
effect of forming equivalence classes of Gaussians that will be
transformed into homogeneous regions after being subjected
to sources of variability.

A top-down binary splitting algorithm is used to cluster
the mean vectors of Gaussians according to their similarity.
The algorithm is similar to that used for regression class
tree construction in MLLR adaptation [23]. In this clustering
algorithm, all Gaussians are initially placed in the root node.
Nodes are then iteratively split at each iteration to maximize
a local maximum likelihood splitting gain criterion. The log
likelihood score of a node, c, which is represented using a
single global mean \( \bar{\mu}_c \) and covariance \( \Sigma_c \), is defined as the
sum of the log likelihood of the Gaussians assigned to this
node,

\[
L(c) = \sum_{m \in c} \ln \left( N_m \cdot N(\bar{\mu}_m, \bar{\mu}_e, \Sigma_c) \right).
\]  

(11)

In Equation 11, \( N_m \) is the number of observation vectors
associated with Gaussian \( m \) in the HMM and \( \bar{\mu}_m \) is the
mean of the observation vectors assigned to Gaussian \( m \). The
splitting procedure repeats until the desired number of nodes
is achieved. The mean subvectors that fall into a terminal node
will form a subvector equivalence class.

### C. Clustered ML Linear Basis Vector Estimation

The expression for the ML estimate of the clustered basis
vectors given in Equation 8 are obtained here using a variant
of the EM algorithm. The procedure for estimating \( \hat{\ell}(k) \)
is a generalization of that used in [5], [10] to account for the
class tying of the basis subvector means according to the
equivalence classes described above. Differentiating the
auxiliary Q function given in Equation 2 with respect to
\( \hat{\ell}(k, n) \) results in the expression

\[
\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \hat{\ell}(k, n)} = \sum_{m \in \phi^{-1}(n)} \sum_{s} \sum_{t} \lambda_m(t, s) \left( \bar{\sigma}^s - \bar{\mu}_m \right)' \Sigma_m \sigma_m^{-1} \bar{w}_k \tag{12}

\]

where \( n \) is the basis subvector index and \( \phi^{-1}(n) = \{ m : \phi(m) = n \} \).

For the unclustered MLLB, the expression for the \( d \)th
components of the \( m \)th basis subvectors, \( \hat{\ell}(m, d) = (e(1, m, d), \cdots, e(K, m, d)) \) where \( e(k, m, d) \) is the \( d \)th
component of \( \hat{\ell}(k, m) \), can be obtained from Equation 12 as

\[
\hat{\ell}(m, d)' = \sum_s \sum_t \lambda_m(t, s) (\bar{\sigma}^s - \mu_m(d)) (\bar{w}^s)' C_m^{-1} \tag{13}
\]

where

\[
C_m = \sum_s \lambda_m(t, s) (\bar{w}^s)' (\bar{w}^s). \tag{14}
\]

Equation 12 can also be solved for the CMLLB case, where the
equivalence classes \( \phi() \) are non-trivial. If a diagonal subvector
covariance matrix, \( \Sigma_m \), is assumed where \( \sigma_m(d) \) is the \( d \)th
diagonal covariance component, then

\[
\hat{\ell}(n, d)' = \sum_{m \in \phi^{-1}(n)} \sum_t \lambda_m(t, s) \cdot (\bar{\sigma}_t^s - \mu_m(d)) \sigma_m^{-1}(d) (\bar{w}^s)' C_m^{-1}(d) \tag{15}
\]

IV. ISSUES FOR IDENTIFYING ADAPTATION SUBSPACE

This section discusses two issues relating to the estimation
of basis vectors for the CMLLB adaptation approach. The first
issue concerns the need for significant amounts of speaker
specific utterances in the training phase. The second issue
concerns the effect of different initialization strategies for the
iterative EM algorithm that used in basis vector estimation.

#### A. Subspace Representations

All of the speaker space approaches that were summarized
in Section I include a "subspace training" phase that relies on
the availability of at least a moderate number of utterances
per speaker. These speaker specific utterances are used to
form the data vectors which are then used for estimating the
basis vectors that span the desired speaker space. Section III
describes the procedure used for estimating the basis vectors
for CMLLB adaptation as an iterative process where, for
each iteration, the weight vectors are first estimated from the
utterances of each speaker and the basis vectors are then
updated. In the next section, it will be shown that a single
two to five second length adaptation utterance is sufficient for
obtaining significant improvement in ASR performance using
CMLLB adaptation. This suggests that it may not be necessary
to use more than one utterance per speaker in training for
CMLLB based basis vector estimation.

The algorithm for basis vector estimation described in
Section III can be modified so that speaker dependent information
is essentially ignored during training. Instead of performing
training using weight vectors that were computed from mul-
tiple speaker specific utterances as implied by Equation 15,
this training can be performed using weight vectors obtained
from each utterance. This corresponds to removing all speaker
specific information from the training set. Mathematically, this
can be described simply by modifying the interpretation of
the "s" superscript indices associated with the observation
vectors, \( \bar{\sigma}^s \), and the weight vectors, \( \bar{w}^s \), in Equations 7 - 16.
For utterance dependent estimation, these indices correspond
to individual utterances rather than the sets of utterances
associated with particular speakers. As mentioned in Section I,
this is an especially important issue in practical applications.
where there is either insufficient speaker specific training data or the training utterances are not labeled according to speaker.

Having removed speaker dependent information during training, it is not correct to say that the low dimensional subspace that is identified during training strictly characterizes interspeaker variability. If the associated basis vectors are estimated from utterance dependent rather than speaker dependent weight vectors, it is reasonable to assume that this low dimensional subspace in some way describes many sources of intraspeaker variability [24] as well. As a result, this will be referred to as an “utterance based” subspace representation rather than a speaker based subspace representation.

B. Initialization of Linear Basis Vectors

It is well known that the EM algorithm is not guaranteed to converge to a global optimum solution [25]. In many applications, the performance of the final solution can vary considerably depending on the choice of the initial estimates for the model parameters. An initial study was performed here to investigate the effect of parameter initialization in the EM algorithm described in Section III for estimating the basis vectors.

![Log Likelihood score trajectories for MLLB estimation. Basis vectors are initialized in random and by eigenvoices respectively.](image)

Fig. 2. Log Likelihood score trajectories for MLLB estimation. Basis vectors are initialized in random and by eigenvoices respectively.

Fig. 2 displays the log likelihoods obtained after each iteration of the EM algorithm for basis vector estimation using four different combinations of initialization and training strategies. Training was performed using the Resource Management (RM) SI-109 training corpus which is described in Section V. Two initialization strategies for the basis vectors were investigated. In the first strategy, the initial basis vectors were obtained through PCA eigenvoice (E) by following a procedure similar to that used in [6]. In the second initialization strategy, the basis vectors were randomly initialized (R). These initialization strategies were applied to training $K = 10$ basis vectors under scenarios that incorporate speaker dependent information in training (S) and that rely strictly on utterance dependent information (U) as described in the previous subsection.

It was observed that there was a negligible difference in ASR WER obtained as a result of using the different initialization strategies in Fig. 2. Furthermore, the figure also shows that there is little difference in training data log likelihood or rate of convergence among the four different combinations. The consistent WER along with the stable convergence properties suggests the training algorithm is relatively insensitive to the initialization strategy. Hence, random initialization of the basis vectors will be used for training all of the systems described in Section V.

V. EXPERIMENTAL STUDY

This section presents an experimental study to evaluate the performance of the MLLB and CMLLB based adaptation procedures described in Section III on the RM and WSJ tasks. The adaptation scenarios of primary interest are those that require small amounts of adaptation speech which, in this study, will be assumed to be single utterances that are 2 to 5 seconds in length.

A. Adaptation and Training Scenarios

Performance comparisons were made with respect to speaker independent ASR on both the RM and WSJ tasks. The SI HMM training scenarios, baseline system configurations, and WERs are described below for each task. Comparisons were also made on the RM task to eigenvoice based speaker adaptation. The eigenvoice implementation, referred to here as $E_{mllr}$, follows the method proposed by Botterweck for speaker space adaptation of large vocabulary HMMs [6]. In the $E_{mllr}$ training procedure, eigenvectors were computed by performing PCA analysis on SD models which were obtained from block diagonal MLLR adaptation of the SI model using utterances for each of the speakers in the SI-109 training set.

The experiments were conducted under unsupervised and supervised adaptation scenarios. A single utterance based unsupervised adaptation scenario was investigated for both the RM and WSJ tasks. This involved a two-pass decoding strategy for each utterance. A hypothesized transcription was obtained using the SI model during the first pass. A weight vector, $\vec{w}$, was estimated from Equation 4 and used to adapt the SI model as described in Equation 1. The final result was obtained in a second decoding pass using the adapted model. A supervised adaptation scenario was also investigated for the RM task. Transcribed speech utterances of varying length were used for estimating SD weight vectors that were used for adapting SI models.

The rest of this section will discuss the evaluation of four different variants of MLLB based adaptation algorithms. Both the unclustered (MLLB) case and clustered (CMLLB) case, will be considered. Basis vector training scenarios that incorporate speaker dependent information in training (S) and that rely strictly on utterance dependent information (U) as defined in Section IV will also be considered.

B. Adaptation on the RM task

Unsupervised and supervised adaptation was performed on the RM corpus under the following scenario. Acoustic SI HMMs were trained using 3990 utterances from 109 speakers taken from the standard RM SI-109 training set. The basis vectors for the $E_{mllr}$ and the MLLB techniques were also trained from this 109 speaker training set. ASR WER was evaluated
using 1200 utterances from 12 speakers taken from the RM speaker dependent evaluation (SDE) set. For supervised adaptation, adaptation utterances were randomly chosen for each of the 12 test speakers from the speaker dependent development (SDD) set. To ensure statistical robustness of the measured WER, six different adaptation sets were randomly chosen for each speaker and the WER obtained after adaptation to these data sets were averaged. The baseline SI HMMs contained left-to-right 3-state state clustered triphones with 6 mixtures per state for a total of 10,224 Gaussians. The standard RM word-pair grammar language model was used. Feature analysis included 12 mel frequency cepstrum coefficients (MFCCs), normalized log energy, and their first and second difference coefficients for a 39-dimension feature vector. The baseline WER of the SI models on the 1200 utterance test set was 4.91%.

1) Unsupervised adaptation experiments: The ASR WER for the unsupervised adaptation scenario evaluated on the RM test set are displayed in Table I including the performance for baseline SI training, eigenvoice ($E_{\text{mllr}}$) based adaptation, MLLB adaptation, and CMLLB adaptation using 1000 shared subvectors. All systems shown were implemented with subspace dimensionalities equal to $K = 10$ and $K = 20$. For both MLLB and CMLLB, WER for speaker dependent (S) and utterance dependent (U) basis vector training as described in Section V-A are shown.

There are several observations that can be made from Table I. First, it is clear that the $E_{\text{mllr}}$, MLLB, and CMLLB approaches all resulted in significant reduction in WER relative to the SI baseline system. The CMLLB approach resulted in as much as a 14.9% reduction in WER relative to the SI system. Second, the CMLLB approach resulted in a 7% WER reduction with respect to the $E_{\text{mllr}}$ eigenvoice implementation, while the MLLB adaptation showed no advantage over $E_{\text{mllr}}$. It is important to note that the CMLLB adaptation only employed approximately one tenth the number of subvectors that were used for the unclustered MLLB, reducing the effective dimensionally of the basis vectors by an order of magnitude. A third observation is that there is no significant difference in WER for any of the techniques when the subspace dimensionality is increased from 10 to 20. Finally, there is no significant difference in WER between the MLLB/CMLLB approaches that occurs as a result of using speaker dependent (S) or utterance dependent (U) basis vector training. This suggests that there is no need to collect large numbers of utterances for each speaker for the purpose of identifying basis vectors during training. This can be an important issue in applications where training utterances are collected from unidentified users of telecommunications services.

<table>
<thead>
<tr>
<th>SubSpace Dimen.</th>
<th>SI</th>
<th>$E_{\text{mllr}}$</th>
<th>MLLB</th>
<th>CMLLB</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.91</td>
<td>4.53</td>
<td>4.62</td>
<td>4.54</td>
</tr>
<tr>
<td>20</td>
<td>4.91</td>
<td>4.48</td>
<td>4.53</td>
<td>4.51</td>
</tr>
</tbody>
</table>

The observation in Table I that CMLLB adaptation consistently resulted in reduced WER relative to the unclustered MLLB adaptation warrants further analysis. In order to demonstrate that this difference in performance is most likely a result of the tendency of the MLLB approach to “overfit” the training data, the unsupervised adaptation experiments described in Table I were repeated on the RM training data and displayed in Table II. It is not surprising that the training data WER displayed for all systems in Table II are lower than the corresponding testing data WER displayed in Table I. However, the extremely large difference between the training and testing WERs shown for all of the MLLB systems suggests that, at least for the RM task domain, the unclustered MLLB approach suffers from over-fitting. Hence, it is very likely that the improved performance demonstrated by CMLLB adaptation is due to the reduction in over-fitting which is achieved through the use of clustered basis vectors. The statistical significance of the differences in measured WER for the systems displayed in Table I was investigated using the suite of significance tests implemented by NIST [26]. Table III displays the results of applying three significance tests including the matched pair (MP) sentence segment (word error) test, the signed paired (SI) comparison test (speaker word accuracy rate), and the Wilcoxon (WI) signed rank test (speaker word accuracy rate). Each entry in the table displays results of a pair-wise significance test of two systems. If two systems are statistically different at a 5% level of significance for a given test, the system with the lower WER will appear in the table entry for that test. If the systems are not judged to be significantly different by a given test, the entry will contain “same” for that test. The subspace dimensionality, $K$, of $E_{\text{mllr}}$ and CMLLB in this test were all equal to 20. Table III shows that the WER differences between the speaker based CMLLB system and both the baseline SI and $E_{\text{mllr}}$ systems given in Table I were all statistically significant according to all of the above tests at a 5% level of significance.

2) Supervised adaptation experiments: The speaker space adaptation approaches described in the previous section were evaluated on the RM task under a supervised adaptation...
TABLE IV
WER for supervised adaptation on the Resource Management (RM) SDE test set.

<table>
<thead>
<tr>
<th>Adapt. Ut.</th>
<th>MLLR</th>
<th>$E_{mllr}$</th>
<th>MLLB</th>
<th>CMLLB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>U</td>
<td>S</td>
<td>U</td>
</tr>
<tr>
<td>1</td>
<td>4.91</td>
<td>4.39</td>
<td>4.67</td>
<td>4.39</td>
</tr>
<tr>
<td>2</td>
<td>4.86</td>
<td>4.39</td>
<td>4.58</td>
<td>4.50</td>
</tr>
<tr>
<td>3</td>
<td>4.44</td>
<td>4.32</td>
<td>4.55</td>
<td>4.44</td>
</tr>
</tbody>
</table>

The effect of varying length adaptation utterances on performance is investigated here and this class of adaptation approaches is compared with MLLR adaptation.

Table IV displays the WER obtained using supervised adaptation with from 1 to 3 adaptation utterances. A subspace dimensionality of $K = 20$ was used for all systems. It is clear that CMLLB outperformed all other systems in Table IV. It is also observed that there was no significant difference for the CMLLB system between the case where speaker dependent information was used for training basis vectors (S) and when it was not used (U). However, there was no significant reduction in WER for the speaker-space based approaches as additional adaptation utterances were used. Table IV also displays the WER for block diagonal maximum likelihood linear regression (MLLR) based adaptation. While it was found that MLLR outperformed all of the subspace based techniques when greater than 4 adaptation utterances were used, MLLR adaptation was inferior to CMLLB for all of the adaptation scenarios shown in the table. Hence, while WER for MLLR adaptation continues to decrease as the number of adaptation utterances increases, the WER of the speaker space approaches appears to have saturated after a single adaptation utterance.

Since Table IV suggests that the supervised adaptation performance for the subspace methods has saturated after a single two to five second length adaptation utterance, an additional study was performed using shorter adaptation utterances to further investigate the effect of adaptation data on performance. Short speech segments drawn from utterances in the RM SDD set were used for supervised CMLLB and $E_{mllr}$ eigenvoice adaptation. Fig. 3 displays the WER for the utterance based CMLLB and eigenvoice adaptation approaches when the length of adaptation data was varied from 0.25 to 2.5 seconds. Subspace dimensionalities of $K = 10$ and 20 were used for both the CMLLB and $E_{mllr}$ systems. It is clear from the figure that the performance of the CMLLB adaptation begins to show significant improvement over the baseline SI system with only 0.5 seconds of adaptation data. The improvement of $E_{mllr}$ adaptation over the baseline become significant once approximately 1.0 second of adaptation speech.

From the results given in Table IV and Fig. 3, one may conclude that the CMLLB approach can perform well in a supervised adaptation scenario when as little as a single word is used for adaptation. However, it is also apparent that none of the subspace based adaptation approaches evaluated on this task are able to benefit from the availability of more than a few seconds of adaptation speech.

C. WSJ evaluation

The CMLLB approach to model adaptation was shown in the previous section to be effective for the RM task where the HMM acoustic model typically consists of approximately 10,000 Gaussians. This section evaluates the behavior of CMLLB adaptation on the large vocabulary WSJ task. A 5,000 word vocabulary WSJ task was evaluated with an acoustic model consisting of over 200,000 Gaussians. The baseline SI system for the WSJ task was configured as follows. The SI HMM’s were trained using 107,937 utterances from 988 speakers contained in the WSJ0, WSJ1, and TIMIT data sets. The basis vectors for the MLLB/CMLLB techniques were trained using the same training sets as the acoustic model. The ASR WER for the single utterance based unsupervised adaptation scenario described in Section V-A was evaluated using the Nov92 test set, which contains 330 utterances. The baseline SI HMM’s contained left-to-right 3-state state clustered triphone models with 16 mixtures per state for a total of 214,256 Gaussian densities. Recognition was performed using a 5k word vocabulary and the WSJ 5k bigram language model. Similar $E_{mllr}$ experiments were not included in the WSJ task due to the difficulty to train speaker dependent models and employ PCA to identify speaker subspace with such huge dimensionality ($39 \times 214,256$).

The ASR WER using unsupervised adaptation for the MLLB adaptation and CMLLB adaptation where each basis vector contains 10,000 shared subvectors are displayed in Table V. Both speaker dependent (S) and utterance dependent (U) basis vector training as described in Section V-A are shown. Subspace dimensionalities of $K = 10$ and 20 were used for all of the adaptation experiments. It is clear from Table V that the unclustered MLLB basis adaptation did not provide a significant reduction in WER relative to the SI system. However, CMLLB adaptation resulted in a WER reduction of 13.9%. The difference in WER evaluated on the WSJ test set between the speaker based CMLLB(S) system and the SI baseline shown in Table V was found to be statistically significant at a 5% level of significance according to the ”MP”, ”SI”, and ”WI” tests. The difference in WER obtained for all systems when increasing the subspace dimension, $K$, from 10 to 20 was not found to be statistically significant.

It is interesting to investigate how the relationship between
An attempt was made here to apply SAT in the context of the CMLLB speaker space based adaptation approach. The procedure for obtaining a more canonical model involves two steps. First, speaker dependent weight vectors, \( \hat{\omega}_s \), are estimated for each speaker, \( s \), in the training corpus and the observation bias vectors for that speaker are computed as follows
\[
\Delta \hat{\omega}_t = \hat{\omega}_t - \mu_m \lambda^s \sum_k w_k \hat{e}_k(m(t)),
\]
where \( m(t) \) is the mixture index assigned to observation vector \( \hat{\omega}_t \) by Viterbi alignment with the SI HMM. Equation 17 corresponds to removing speaker dependent variation from the training vectors. It is clear from Equation 7 that attaching the speaker dependent variation to the SI model to form the SD model as shown in Equation 17 is equivalent to removing speaker variation in the corresponding speech data. In the second step, it can be shown that the SAT subvector means, \( \hat{\mu}_m \), and variances, \( \hat{\Sigma}_m \), are obtained from the observation bias vectors as
\[
\hat{\mu}_m = \frac{\sum_s \sum_t \lambda^s_m(t,s) \Delta \hat{\omega}_t^s + \mu_m}{\sum_s \sum_t \lambda^s_m(t,s)}
\]
\[
\hat{\Sigma}_m = \frac{\sum_s \sum_t \lambda^s_m(t,s) \Delta \hat{\omega}_t^s \Delta \hat{\omega}_t^s'}{\sum_s \sum_t \lambda^s_m(t,s)}
\]
where \( \lambda^s_m(t,s) \) is the occupation probability of the \( n \)th Gaussian density for observation vector \( \hat{\omega}_t \) using speaker dependent HMMs. Hence, SAT training for CMLLB involves estimating speaker dependent weight vectors from training utterances, using the weight vectors to update the observation bias vectors, and then reestimating the model means and covariances with the observation bias vectors.

No significant reduction in ASR WER was observed when performing CMLLB adaptation after SAT using the above strategy. One can speculate on why CMLLB based SAT was found to be ineffective even though MLLR based SAT has consistently been found to result in significant reductions in WER. The answer is most likely in the superior adaptation performance that is obtained by MLLR adaptation when enough adaptation utterances are available. MLLR based SAT is generally applied in training scenarios where on the order of a minute of speech is available for each training speaker. With that amount of data per speaker to estimate the speaker dependent MLLR regression matrices in SAT, it is possible to considerably reduce the impact of speaker variability in training [27] [28]. However, while speaker space based adaptation techniques perform far better than MLLR for very short utterances, they have far less impact than MLLR when on the order of a minute of speech is available. Hence, in the limit of very large speaker dependent data sets, CMLLB based transformations have nowhere near the power of MLLR transformations to reduce the effects of speaker variability. This may be the main source of the poor performance of CMLLB based SAT relative to MLLR based SAT.

### Table V
WER for unsupervised adaptation on the Wall Street Journal (WSJ) Nov92 test set

<table>
<thead>
<tr>
<th>SubSpace Dimen.</th>
<th>SI</th>
<th>MLLB</th>
<th>CMLLB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>U</td>
<td>S</td>
</tr>
<tr>
<td>10</td>
<td>5.17</td>
<td>5.12</td>
<td>5.08</td>
</tr>
<tr>
<td>20</td>
<td>5.17</td>
<td>5.19</td>
<td>5.10</td>
</tr>
</tbody>
</table>

### Table VI
WER for unsupervised adaptation on the Wall Street Journal (WSJ) Nov92 test set as the number of clustered subvectors is varied from 500 to 16000 for utterance based (U) and speaker based (S) basis vector training.

<table>
<thead>
<tr>
<th>SubSpace Dimen.</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>6000</th>
<th>10000</th>
<th>16000</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>5.23</td>
<td>4.86</td>
<td>4.61</td>
<td>4.67</td>
<td>4.52</td>
<td>4.61</td>
</tr>
<tr>
<td>S</td>
<td>5.31</td>
<td>5.27</td>
<td>4.73</td>
<td>4.71</td>
<td>4.63</td>
<td>4.67</td>
</tr>
</tbody>
</table>

the physical basis vector dimensionality, \( N \), and the logical basis vector dimensionality, \( M \), affects WER in CMLLB adaptation. With 10,000 subvectors, the CMLLB implementation shown in Table V reduced the basis vector dimensionality by a factor of 20. Table VI displays the WER for the CMLLB adaptation with subspace dimensionality of \( K = 10 \) as the number of clustered subvectors is varied over a range from 500 to 16,000. With the original supervector consisting of approximately 200,000 Gaussians, this corresponds to reducing the basis vector dimensionality through clustering by a factor ranging from as much as 400 to as little as approximately 12. The first row in Table VI displays the WER for the cases where utterance-based basis vector training was used, and the second row displays WER for speaker-based basis vector training.

Of course, the proper ratio of the number of physical subvectors to the number of logical subvectors depends on the amount of training data available and the nature of the task domain. It is clear from Table VI that the minimum WER for both speaker-dependent and utterance-dependent basis vector training was obtained using 10,000 clustered subvectors. It is also true that in both cases, significant reductions in WER with respect to the original SI baseline system were obtained when only 2,000 clustered subvectors were used. This amounts to a potential memory reduction of over two orders of magnitude with respect to the original unclustered MLLB adaptation for the storage of basis vectors.

### VI. Discussion

The general technique of speaker adaptive training (SAT) has been applied to the problem of removing speaker dependent variability in acoustic model training. It has been demonstrated that it is possible to create a more “canonical” speaker independent HMM by adapting either the model parameters [10], [27] or the training utterances [28] as part of a procedure to reduce the variability that is characterized by the model. When this new model is used as part of an adaptation scenario in testing, more efficient adaptation behavior has been observed and significant reductions in WER have been obtained. SAT has been applied in the context of MLLR adaptation [27], constrained MLLR adaptation [28], and cluster adaptive training (CAT) [10].
VII. CONCLUSIONS AND FUTURE WORK

A subspace approach to speaker adaptation relying on maximum likelihood estimation of clustered linear basis vectors, CMLLB, has been presented. The approach was shown to be robust with respect to sparsity of training data and also efficient in terms of its ability to achieve good adaptation performance with extremely small amounts of adaptation data. It was applied to single utterance based unsupervised speaker adaptation on the RM and WSJ tasks. WER reductions of 14.9% and 13.9% relative to SI training were obtained for RM and WSJ respectively. Furthermore, it was found that the memory required for storing linear basis was reduced by over an order of magnitude with respect to unclustered subspace adaptation methods resulting in a savings of well over 100 Mbytes for the WSJ task.

A variant of CMLLB has also shown that it requires no speaker specific information for basis vector training performed as well as standard scenarios where a large number of utterances are required from each training speaker. This result may be important when considering the significance of the "speaker subspaces" that are identified by Eigenvoice and CAT approaches. While the subspaces that are being identified by these speaker space approaches have been shown to be effective for improving ASR adaptation results, it is not clear that the subspaces themselves specifically represent speaker dependent variation or something more general. A similar conclusion drove a recent implementation of RSW [8], [9]. Estimation of basis vectors is not performed at all prior to adaptation in this implementation, and speaker dependent supervectors are obtained simply by linear combination of speaker models that are found to be "close" to the adaptation utterance. Together, the success of these two techniques that rely on very different notions of subspaces for adaptation may lead to other more general subspace based adaptation approaches.

A natural extension of the subspace based adaptation approaches that were addressed in this paper is to explore the applications of nonlinear methods. The performance of the subspace based approaches is limited by the fact that the adapted mean vectors are formed by interpolating linear basis vectors. Non-linear kernels have been applied in eigenvoice based speaker adaptation [14], [15], and were found to compare favorably with eigenvoice methods implemented using a linear subspace. Nonlinear manifold based approaches have also been successfully applied in image understanding applications [29], [30]. It is reasonable to expect that representing sources of speaker variability on a non-linear manifold may also result in improved speaker adaptation performance. Exploiting non-linear methods while maintaining the efficiency and ease of implementation of the techniques described in this paper will be addressed as areas of future work.

APPENDIX I

This appendix provides the derivation of the expression for the clustered maximum likelihood linear basis functions given in Equations 15 and 16. The derivation begins with the formulation of the auxiliary Q function

\[ Q(\Lambda, \hat{\Lambda}) = - \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t - \hat{\mu}_m)^\top \Sigma_m^{-1}(\hat{o}_t - \hat{\mu}_m) \]

\[ \hat{\mu}_m = \hat{\mu} + \sum_{k=1}^{K} w_k \hat{e}(k, m) \]

Differentiating the auxiliary Q function with respect to \( \hat{e}(k, n) \) and setting to zero results in the expression,

\[ \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t - \hat{\mu}_m)^\top \Sigma_m^{-1} w_k^s = 0 \]

If only one component in subvector is considered and the covariance matrix \( \Sigma_m \) is supposed to be diagonal matrix with diagonal elements \( \{\sigma(1), \sigma(2), \ldots, \sigma(D)\} \), then we get,

\[ \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t^s - \mu_m(d)) - \sum_{k'} w_k^s \hat{e}(k', n, d) \sigma_m^{-1}(d) w_k^s = 0 \]

\[ \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t^s - \mu_m(d)) - (\hat{u}^s)^\top \hat{e}(\cdot, n, d) \sigma_m^{-1}(d) w_k^s = 0 \]  (20)

After vectorizing Equation 20 for all \( k = 1, \ldots, K \), the above equation could be expressed as,

\[ \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t^s - \mu_m(d)) - (\hat{u}^s)^\top \hat{e}(\cdot, n, d) \sigma_m^{-1}(d) (\hat{u}^s)^\top = 0 \]

It follows that,

\[ \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s)(\hat{o}_t^s - \mu_m(d)) \sigma_m^{-1}(d) (\hat{u}^s)^\top = \hat{e}(\cdot, n, d)^\top \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s) (\hat{u}^s) \sigma_m^{-1}(d) (\hat{u}^s)^\top \]

So the clustered basis vectors that maximize the auxiliary Q function could be computed as,

\[ \hat{e}(\cdot, n, d)^\top = \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s) \cdot \]

\[ (\hat{o}_t^s - \mu_m(d)) \sigma_m^{-1}(d) (\hat{u}^s)^\top C_n^{-1}(d), \]

\[ C_n(d) = \sum_{m \in \phi^{-1}(n)} \sum_{s=1}^{S} \sum_{t=1}^{T} \lambda_m(t, s) (\hat{u}^s) \sigma_m^{-1}(d) (\hat{u}^s)^\top \]

where \( C_n(d) \) is a \( K \times K \) subvector component dependent matrix.
ACKNOWLEDGMENT

The authors would like to thank Parya Momayyez for training the WSJ baseline system. We would also like to thank three anonymous reviewers, whose comments helped to make this paper better.

REFERENCES