Speaker Clustering and Transformation for Speaker Adaptation in Speech Recognition Systems

Mukund Padmanabhan, *Member, IEEE,* Lalit R. Bahl, *Fellow, IEEE,*
David Nahamoo, *Member, IEEE,* and Michael A. Picheny, *Member, IEEE*

***Abstract*— A speaker adaptation strategy is described that is based on finding a subset of speakers, from the training set, who are acoustically close to the test speaker, and using only the data from these speakers (rather than the complete training corpus) to reestimate the system parameters. Further, a linear transformation is computed for every one of the selected training speakers to better map the training speaker's data to the test speaker's acoustic space. Finally, the system parameters (Gaussian means) are reestimated specifically for the test speaker using the transformed data from the selected training speakers. Experiments showed that this scheme is capable of providing an 18% relative improvement in the error rate on a large-vocabulary task with the use of as little as three sentences of adaptation data.**

***Index Terms*— Data transformation, speaker adaptation, speaker clustering.**

I. Introduction

IN THE LAST few years, several advances have been made in improving the error rate of continuous-speech-recognition systems [1]. For instance, the best word-error rates on test data drawn from the *Wall Street Journal* (*WSJ)* data base—as reported by different participants in the *WSJ* task [1]—hover in the neighborhood of 7–8% for large-vocabulary speaker-independent systems. Though this represents a reasonable level of performance on this particular test data, there is still scope for further improvement. One way to improve the performance of these systems is to make the system parameters speaker dependent. However, large-vocabulary systems tend to have a large number of parameters, and in order to robustly estimate these parameters, a large amount of training data is needed. This implies that the test speaker will have to furnish a large amount of data to specifically train the system to his/her speech. This is usually not a practical solution. Consequently, there is increasing interest in speaker adaptation techniques that require only a small amount of data from the test speaker. This data is used to move the parameters of the speaker-independent system toward speaker-dependent values.

In this paper, we present a speaker adaptation method that is based on finding a cluster of speakers who are acoustically "close" to the test speaker, then individually transforming each of these training speakers' data to bring it closer to the test speaker's acoustic space, and using this transformed data to estimate the model parameters.

Manuscript received February 17, 1996; revised February 14, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mazin Rahim.

The authors are with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: mukund@watson.ibm.com).

Publisher Item Identifier S 1063-6676(98)00589-6.

II. Technical Background

A. System Overview

We will first briefly describe the IBM large-vocabulary speech recognition system. Essential aspects of the system used in the experiments here have been described earlier [2]–[4]; however, we will summarize the main features here.

1) Signal Processing: A 60-dimensional (60-D) feature vector is extracted from the input waveform at regular intervals of 10 ms [4]. The processing involves 1) computing 24-band mel cepstra using a 25 ms window for the fast Fourier transform (FFT), 2) splicing together the cepstra from the adjacent s frames on either side of the current frame (typically $s = 4$, resulting in a 216-dimensional vector), and 3) applying a linear transformation that brings the dimensionality of the vector down to 60-D.

The linear transformatiom mentioned above is actually a composition of two linear transformations derived from the training data using linear discriminant analysis [4]. In the first step, the linear discriminants of the unspliced 24-dimensional (24-D) cepstra are obtained, and applied on the cepstra. There is no change in dimensionality at this stage. The second step of the technique attempts to capture the dynamics of speech in this transformed 24-D space. This is done independently for each dimension, d, of the transformed space. The dth component of the transformed cepstra across $2s + 1$ frames are taken and linear discriminants are obtained to maximally separate subphonetic classes on the basis of this $(2s + 1)$-dimensional vector. Subsequently, the 60 most discriminative projections are chosen and put together with the first rotation to give the final composite transformation.

2) Acoustic Models: Words are represented as sequences of phones. Each phone is further divided into three subphonetic units, which correspond roughly to the beginning, middle, and end of each phone. The system uses context-dependent hidden Markov model (HMM) acoustic models for these subphonetic units. For each subphonetic unit, a decision tree is constructed from the training data [2]. Each leaf of the tree corresponds to a different set of contexts. The acoustic observations that characterize the training data at each leaf are modeled as a mixture of 60-D Gaussian probability density functions (pdf's), with diagonal covariance matrices. The HMM's used to model the leaves are simple two-state models, with a self-loop and

a forward transition. For an observed acoustic vector, we compute the pdf value at each leaf. However, the pdf values are not used directly. In order to obtain a more robust model, we compute the rank of each leaf by sorting the entire set of pdf values. The output distribution of each HMM is modeled as a discrete distribution on the ranks [3]. The system used in this paper had approximately 6000 leaves and 17 000 Gaussians.

3) Training Data: The training corpus for the *WSJ* task consists of 100–200 utterances from each of 284 speakers. The total corpus size is about 35 000 utterances. A transcription of each utterance at the word level is available. If a word has multiple possible pronunciations, we refine the transcription to indicate which particular pronunciation was used in that utterance. We also indicate the presence of pauses between words. Both these modifications to the original word-level script are done automatically. Once the modified transcription is available, it is easily turned into the corresponding sequence of leaves using the Viterbi alignment procedure, and each acoustic vector from an utterance can be identified with the leaf it belongs to.

B. Review of Adaptation Techniques

Some adaptation schemes that have been proposed recently include transformation methods [5]–[7], maximum *a posteriori* (MAP) estimation [8], [9], etc. In [5], the speaker-independent system is transformed to come closer to the test speakers acoustics by applying a linear transformation on the means of the speaker-independent Gaussians. The transformation is computed so as to maximize the likelihood of the test speaker's adaptation data. The scheme used in [6] is similar (the transformations are however constrained to be diagonal)—here, the assumption is made that the acoustic space of the test speaker and the training data are related by a linear transformation, and the model parameters are reestimated for the test speaker by applying this transformation on the means and covariance matrices of the speaker-independent system. Another related scheme that applies a nonlinear transformation on the training data, in order to map it to the test speakers space, is the metamorphic transformation of [7]. In contrast to these transformation schemes, [8] and [9] attempt to obtain a Bayesian estimate of the model parameters from the limited amount of adaptation data available from the test speaker. These schemes assume a prior distribution on the model parameters, that leads to a very simple adaptation process.

In contrast to the above schemes, the adaptation scheme described here is based on the fact that the training data contains a number of training speakers, some of whom are closer, acoustically, to the test speaker, than the others [10].[1] If the model parameters are reestimated from the subset of training speakers who are acoustically close to the test speaker, they should be reasonably close to the speaker-dependent parameters that would be obtained by training on large amounts of data from the test speaker (if such data were available) [13], [14].[2]

A further improvement on speaker-clustering can be obtained if the acoustic space of each of these training speakers is transformed to come even closer to the test speaker, to minimize the mismatch between the test and training data. This may be done by using linear [5] or nonlinear [7] techniques; in this paper, for reasons of simplicity, we have opted to use the maximum likelihood linear regression (MLLR) technique of [5].

The adaptation scheme is described in more detail in the following section, and is shown to be capable of giving reasonable improvements in performance with a very little amount of adaptation data. The notation used in the rest of the paper is as follows: underlining will be used to represent a column vector, and double underlining will be used to represent a matrix.

III. The Adaptation Procedure

The adaptation procedure is summarized in Fig. 1, and comprises the following steps. First, construct an acoustic model for each of the speakers in the training corpus. Next, using the adaptation data to characterize the test speaker, find a subset of the training speakers who are acoustically "close" to the test speaker. Then, compute a linear transform using the MLLR technique of [5] to map the acoustic space of each selected training speaker closer to the test speaker's acoustic space. Finally, reestimate the Gaussians of the speaker-independent model using the transformed data from the selected training speakers. The various steps in the adaptation procedure are described next.

A. Models for the Training Speakers

For the purpose of speaker clustering, it is necessary to obtain an acoustic characterization of each the 284 training speakers in order to determine which training speakers are close to the test speaker. We chose to model the acoustic characteristics of each speaker by a single Gaussian per leaf (6000 Gaussians).[3] However, the 100–200 utterances that are available from each training speaker are not sufficient to obtain robust estimates of the parameters of the speaker-dependent models. Consequently, we used Bayesian adaptation techniques [8] to smooth each speaker-dependent model with a speaker-independent model.

For purposes of notation, let L denote the total number of leaves, d denote the dimension of the acoustic features, and $\underline{\mu}_i^{\text{ind}}, \underline{\underline{\Lambda}}_i^{\text{ind}} i = 1, \cdots, L$ denote the parameters of a speaker-independent acoustic model that models each leaf with a single diagonal Gaussian ($\underline{\mu}_i^{\text{ind}}$ is a d-dimensional vector and $\underline{\underline{\Lambda}}_i^{\text{ind}}$ is a dxd diagonal matrix); further, let the kth training speaker be parametrized by $\underline{\mu}_i^k, \underline{\underline{\Lambda}}_i^k, i = 1, \cdots, L$, with $\underline{\underline{\Lambda}}_i^k$ being diagonal. The MAP reestimation strategy of [8] assumes a prior distribution $p(\theta)$, on the parameters being estimated, $\theta = (\underline{\mu}_i^k, \underline{\underline{\Lambda}}_i^k)$, and attempts to find

$$\theta_{\text{MAP}} = \arg \max_{\theta} p(\theta) p(\underline{y}_1^T / \theta) \tag{1}$$

[1] Some similar ideas have recently been reported in [11] and [12].

[2] The simplest implementation of such a clustering strategy would be gender-dependent processing.

[3] Because of storage constraints, the training speaker models (6000 Gaussians) are much smaller than the speaker-independent and speaker-adapted systems (17000 Gaussians).

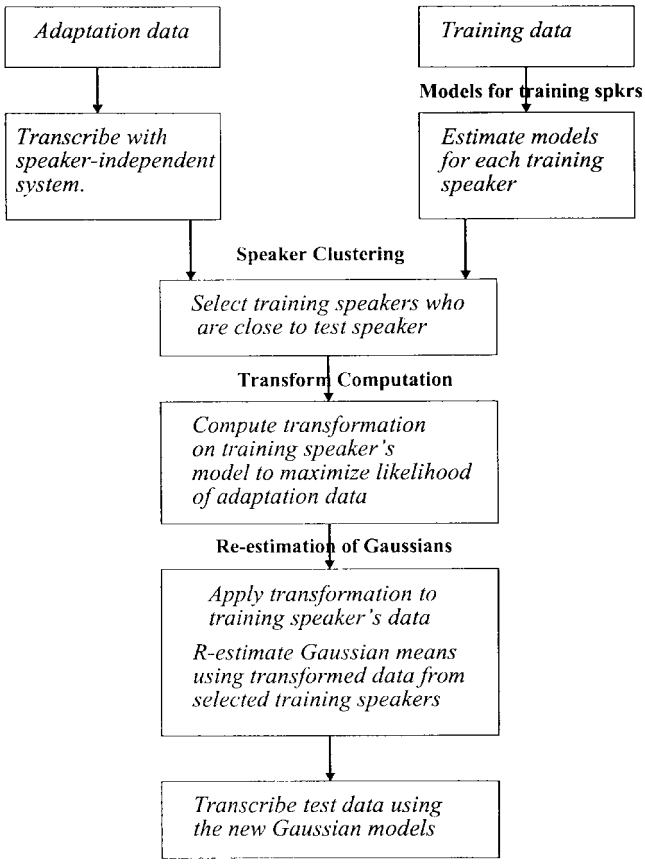

Fig. 1. Adaptation procedure.

where $\underline{y}_1^T$ is the training data from the kth speaker. In [8], it was shown that the choice of a normal-Wishart density for the the prior distribution on the Gaussian parameters, $p(\theta)$, resulted in a convenient estimation strategy. Consequently, choosing the prior distribution to be of the form

$$p(\theta) = |\underline{\underline{\Lambda}}_i^{\text{ind}^{-1}}|^{\tau_i} \exp\left[-\frac{\tau_i}{2}\left(\underline{\mu}_i^k - \underline{\mu}_i^{\text{ind}}\right)^T \underline{\underline{\Lambda}}_i^{k^{-1}}\left(\underline{\mu}_i^k - \underline{\mu}_i^{\text{ind}}\right)\right] \cdot \exp\left[-\frac{1}{2}\,\text{tr}\left(\underline{\underline{\Lambda}}_i^{\text{ind}}\underline{\underline{\Lambda}}_i^{k^{-1}}\right)\right] \quad (2)$$

leads to the reestimation formulae

$$\underline{\mu}_i^k = \frac{\underline{\eta}_i + \tau_i \underline{\mu}_i^{\text{ind}}}{c_i + \tau_i} \quad (3)$$

$$\underline{\underline{\Lambda}}_i^k = \frac{1}{c_i + \tau_i}[\underline{\underline{\gamma}}_i + \tau_i[\underline{\underline{\Lambda}}_i^{\text{ind}} + \underline{\mu}_i^{\text{ind}}\underline{\mu}_i^{\text{init},T}]] - \underline{\mu}_i^k \underline{\mu}_i^{k,T} \quad (4)$$

where

$$c_i = \sum_t c_i(t) \quad \underline{\eta}_i = \sum_t c_i(t)\underline{y}_t$$
$$\underline{\underline{\gamma}}_i = \sum_t c_i(t)\underline{y}_t\underline{y}_t^T \quad \tau_i = \text{const.} \quad (5)$$

Here, $c_i(t)$ is the *a posteriori* probability of the leaf i at time t, conditioned on all acoustic observations $\underline{y}_1^T$, and the terms $c_i(t), \underline{\eta}_i, \underline{\underline{\gamma}}_i$ are usually referred to as the E-M counts. The parameter τ_i in the expression for the prior distribution (2) is usually chosen to be a constant.

B. Speaker Clustering

The next step in the adaptation procedure is to find a subset of the training speakers who are closest to the test speaker. The adaptation data from the test speaker is first decoded using a speaker-independent system (with 17 000 Gaussians) in order to obtain a transcription. Subsequently, the data is Viterbi aligned against the transcription and each acoustic observation is tagged with a leaf id. The acoustic likelihood of the adaptation data, conditioned on this alignment, is then computed using each training speaker's model, and the training speakers are ranked in the order of this likelihood. The top N speakers are then picked as being acoustically close to the test speaker.

C. Transform Computation

A transformation to bring a training speaker's data closer to the test speaker's acoustic space may be computed in several ways [5]–[7]. For reasons of simplicity, we have opted to use the MLLR technique of [5]. We will briefly summarize this procedure next. Recall that we have already obtained a transcription of the adaptation data using a speaker-independent system with 17 000 Gaussians. Using this transcription and the speaker-independent model, it is possible to compute the posterior probability, $c_i(t)$, of the ith leaf at time t, conditioned on all the acoustic observations in the adaptation data. Unlike the MLLR technique, however, $c_i(t)$ is not obtained using the model for the kth training speaker, but is obtained using the gender-independent model.[4]

We will assume that a linear transformation, $\underline{\underline{A}}^k$, is applied to the means of the training speaker's model, $\underline{\mu}_i^k$, and compute the transformation so as to maximize the likelihood of the adaptation data, given the training speaker's model. This is equivalent to minimizing the following objective function [5]:

$$\sum_{i,t} c_i(t)[(\underline{x}_t - \underline{\underline{A}}^k\underline{\hat{\mu}}^k{}_i)^T \underline{\underline{\Lambda}}^{k^{-1}}_i(\underline{x}_t - \underline{\underline{A}}^k\underline{\hat{\mu}}^k{}_i) + \log(|\underline{\underline{\Lambda}}^k{}_i|)]. \quad (6)$$

Here, $\underline{\underline{A}}^k$ is a $(d)x(d+1)$ matrix, and $\underline{\hat{\mu}}^k{}_i$ is a $(d+1)x1$ vector obtained from $\underline{\mu}^k{}_i$ as $(\underline{\hat{\mu}}^k{}_i)^T = [(\underline{\mu}^k{}_i)^T 1]^T$. The reestimation formulae for $\underline{\underline{A}}^k$ are identical to those in [5] and will not be repeated here.

In the above development, it was assumed that the same matrix $\underline{\underline{A}}^k$ was applied to all means. However, if sufficient data is available, it is possible to compute several transformations, with different transformations being applied to disjoint clusters of leaves. The clusters can be obtained based on the acoustic similarity of the leaves using a bottom-up procedure as in [5].

[4]The reason for doing this was that the models for the training speakers are very crude (only 6000 Gaussians); consequently, one could expect the alignment of states produced by using these models to be much poorer than for the case where the larger gender-independent model is used. The expectation maximization (EM) algorithm typically gives the posterior probability, $c_j(t)$ of the jth Gaussian, at time t, conditioned on all the acoustic observations. By summing these probabilities over the Gaussians that model a leaf, i, the posterior probability of the leaf at time t, $c_i(t)$, conditioned on the acoustic observations can be obtained.

D. Reestimation of the Gaussians

Once the transformations have been computed, one possibility is to accumulate the transformed model means of the selected training speakers to obtain the means of a speaker-adapted system. However, as the training speaker models used only 6000 Gaussians, this would result in a speaker-adapted system with only 6000 Gaussians. This is not desirable, as our original objective was to obtain a speaker-adapted version of a much larger system that had 17 000 Gaussians. Consequently, though the above formulation computed a linear transformation on the training speaker models, in the final stage of the adaptation procedure these transformations will be applied to the training data rather than to the models. The rationale for this comes from the fact that the transformed means of the training speaker's model can be obtained either by applying a linear transformation on the original means of the speaker, or by applying the same transformation on the training data, and then estimating the means from the transformed data. The transformed data is then used to reestimate the means of the larger 17 000 Gaussian system. For the case where multiple transformations are applied to a training speaker, it is necessary to know what leaf (context-dependent subphonetic state) an acoustic observation corresponds to, in order to apply the appropriate transformation. This information is obtained from an existing Viterbi alignment of the training data.

The Gaussian means are reestimated from the training data of the selected speakers using the reestimation formulae given below. Let

- $\underline{x}_t^k$ be the tth acoustic observation from the kth speaker;
- l_t be the leaf (context-dependent subphonetic state) corresponding to the acoustic observation $\underline{x}_t^k$;
- $\underline{\mu}_{i,j}$ be the jth Gaussian of the ith leaf of the speaker-independent system
- $c_{l_t,j}^k(t)$ be the posteriori probability of the jth Gaussian of the leaf l_t conditioned on the current acoustic observation, and the alignment, i.e., summing $c_j^k(t)$ over all the Gaussians that model the current leaf l_t equals one;
- $\underline{\underline{A}}_i^k$ be the transformation corresponding to the ith leaf of the kth speaker.

Then, the mean of the jth Gaussians modeling leaf i may be reestimated as

$$\underline{\mu}_{i,j}^{\text{adapted}} = \frac{\sum_k \underline{\underline{A}}_i^k \left[\sum_t c_{i,j}^k(t) \underline{x}_t^k \right]}{\sum_k \sum_t c_{i,j}^k(t)}. \tag{7}$$

IV. Experimental Results

This section summarizes the results of various experiments that were conducted to evaluate the speaker-adaptation algorithm. One set of test data (Test 1) comprised of 20 sentences from ten speakers (five males, five females). The test speakers were drawn from the *WSJ* SI-37 training data base.[5] The system used to transcribe the test data had 6000 leaves, and 17 000 Gaussians modeling the leaves. The language model used was the official 20 K language model that was provided by the National Institute of Standards and Technology [1] for the November 1994 Advanced Research Projects Agency (ARPA) evaluation, which represents a 97.6% coverage of the test vocabulary. The adaptation data for each test speaker ranged from 3–30 sentences (20–220 s). The other set of test data (Test 2) comprised about 15 sentences (150 s) from each of 20 speakers from the November 1994 evaluation data of the *WSJ* task. Unsupervised adaptation was used in all cases, i.e., the adaptation data was transcribed with the speaker-independent system, and the transcription used for further processing.

TABLE I
Test 1

	Baseline	N=20	N=30	N=50	N=70	N=142
Error (%)	14.43	14.15	13.47	13.5	13.47	14.15
Rel Impr (%)	NA	1.9	6.6	6.6	6.6	1.9

A. Speaker Selection

The first experiment examines the effect of picking a subset of N training speakers who are close to the test speaker, and reestimating the model parameters from the training data provided by the selected speakers. The adaptation data comprised of three sentences from each speaker (20 ss). The performance of the system is shown in Table I as a function of the number of selected speakers, and is seen to provide, at best, a relative improvement of 6.6%.

An interesting observation may be made at this stage by examining the training speakers who are hypothesized to be close to the test speaker. Fig. 2 shows the distance between a male test speaker and each of the 142 male and 142 female training speakers in the *WSJ* SI-284 corpus. For the sake of ease of interpretation, the distances have been sorted before being plotted, and the distances to the male and female training speakers are plotted separately. It can be seen that if the closest N training speakers were selected, they could include male as well as female speakers.

B. Speaker Selection and Transformation

In this experiment, three sentences ($\approx$ 22 s) from each test speaker were used as the adaptation data, and a global transformation was computed for each test-training speaker pair. Table II shows the error rate as a function of the number of training speakers, N, selected to reestimate the Gaussians of the adapted system. For comparison purposes, the error rate obtained with the MLLR adaptation scheme [5] is also shown. It can be seen that the MLLR scheme of [5] yields a 10.3% improvement over the baseline. In contrast, the best performance of the clustering/transformation technique corresponds to using $N = 50$, and is around 18% better than the baseline system, a relative improvement of 7.7% over the MLLR scheme. The value of $N = 50$ will be used in all subsequent clustering/transformation experiments.

[5] The reason for selecting the test data was that each of the test speakers has around 1200 sentences of data that can be used to estimate a speaker-dependent system, and thus compare its performance with that of the speaker-adapted system.

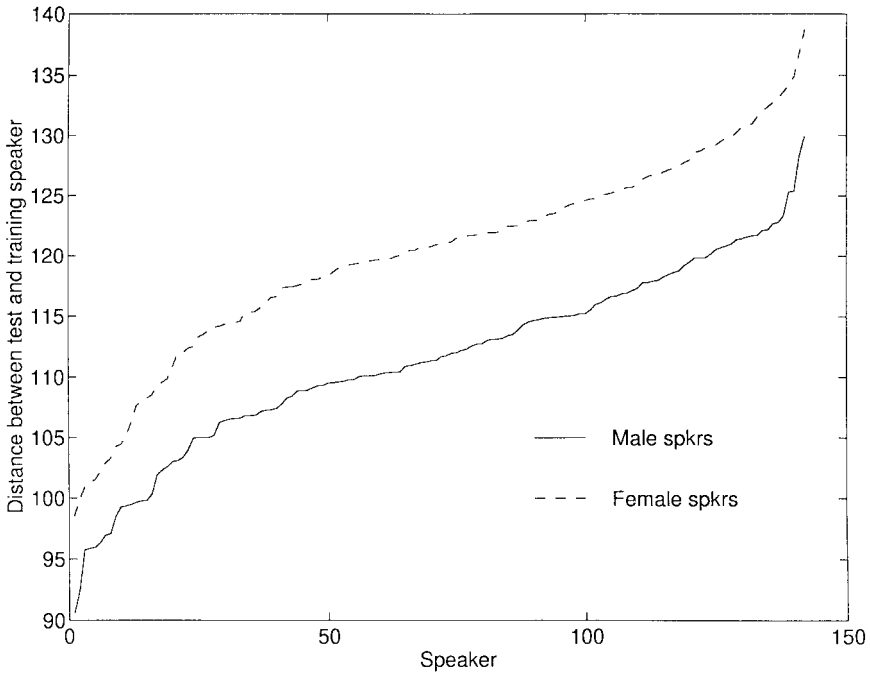

Fig. 2. Distances between a test speaker and the training speakers.

TABLE II
TEST 1

	Baseline	MLLR	N=284	N=142	N=70	N=50	N=30	N=20
Error (%)	14.43	12.95	12.46	12	12.24	11.84	12.06	12.24
Rel Impr (%)	NA	10.26	13.65	16.84	15.18	17.95	16.42	15.18

C. Effect of Increasing Amount of Adaptation Data

In this experiment, we examine the effects of using additional adaptation data, and of using multiple transforms to map each training speaker's data into the test speaker's acoustic space. We conducted two experiments that used three and 15 sentences, respectively, of adaptation data from each test speaker. For the former case, as the amount of adaptation data is very limited, it was only possible to estimate a single global transformation for every test-training speaker pair; for the latter case, there is sufficient data to compute more than one transformation per speaker-pair, with acoustically similar leaves sharing a transformation. The use of the bottom-up procedure mentioned in Section III-C resulted in an average of two transformations being made for each test-training speaker pair. The value of N (number of training speakers selected as being close to the test speaker) was set to 50. The results are shown in Table III. For comparison purposes, the error rates obtained with the MLLR scheme are also shown. It can be seen from the table that, though the performance improves with the use of additional adaptation data, the improvement due to the additional data is quite small; increasing the amount of adaptation data five-fold only increases the relative improvement from 18 to 19.5%.

D. Comparison to Speaker-Dependent Baseline

In this experiment, we compare the performance improvement obtained by the clustering/transformation technique to speaker-dependent results. As mentioned earlier, around 1200 sentences are available for each test speaker, and we used Bayesian adaptation [8] to reestimate the model paramaters for each test speaker from this data. As 1200 sentences represents a fairly large amount of training data, we will assume that the performance of the Bayesian adapted system is very close to speaker-dependent performance. The results obtained with the Bayesian adapted system are summarized in Table IV, along with the results obtained with the clustering/transformation adaptation scheme with three adaptation sentences.

It can be seen from Table IV that the adaptation technique proposed in this paper, with the use of only three adaptation sentences, gives an 18% relative improvement in the error rate; that is more than half the 30% relative improvement that can be obtained by speaker-dependent training with 1200 sentences of speaker-dependent training data.

E. Auto-Adaptation on WSJ Task

Finally, in this experiment, we present results using auto-adaptation on the Test 2 data. As mentioned earlier, this data comprises about 15 sentences ($\approx$ 150 s) from 20 speakers, and represents a standardized data base that was used for the November 1994 evaluation of the *WSJ* task. The test data was first transcribed with the speaker-independent system, and all of this data was used as the adaptation data for reestimating the model paramaters. This resulted in an average of three transformations being made per test-training speaker pair. As

TABLE III
TEST 1

		3 sentences		15 sentences	
	Baseline	MLLR	Clstr/Trans	MLLR	Clstr/Trans
Error (%)	14.43	12.95	11.84	12.33	11.62
Rel Impr (%)	NA	10.26	17.95	14.55	19.47

TABLE IV
TEST 1

	Baseline	Clstr/Trans	Bayesian
Error (%)	14.43	11.84	10.15
Rel Impr (%)	NA	17.95	29.66

TABLE V
TEST 2

	Baseline	MLLR	Clstr/Trans
Error (%)	14.76	12.91	12.25
Rel Impr (%)	NA	12.53	17.00

in earlier experiments, N was set equal to 50. The same data was then redecoded using the adapted model. The results of the experiment are summarized in Table V. The results obtained with the MLLR scheme are also shown in the table. From Table V, it can be seen that the clustering/transformation scheme provides a relative improvement of 17%, which is about 4.5% better than the 12.5% improvement provided by the MLLR scheme.

V. Computational Complexity

The performance improvement of the clustering/transformation technique is obtained, however, at the expense of a large increase in complexity. As mentioned in Section III, the various steps involved in the clustering/transformation scheme during the decoding process are

1) selecting N training speakers who are closest to the test speaker;
2) computing a transformation that maps each training speaker to the test speaker;
3) applying the transformation to either the training speaker models, or the training speakers data to reestimate the model parameters for the test speaker.

The computational cost of selecting the closest training speakers is relatively small, and the major part of the computation is associated with steps 2 and 3. In step 2, a separate linear transformation is computed using the MLLR technique for each test-training speaker pair; hence, the computation in this step is N times that of the MLLR scheme (typically, $N = 50$). The computation required in step 3 depends on whether the transformation is applied on the selected training speaker's models, or on the training speaker's data. In the former case, the computation is negligible, but a penalty is incurred in terms of storage, as full-scale models for all the training speakers have to be stored (284 speakers $\times$ 8 Mbytes per speaker). In the latter case, a computational penalty is incurred because the training data has to be processed again to reestimate the model means.

However, in spite of the increased complexity, there are several tasks, such as the ARPA-sponsored *WSJ* [4] and Hub 4 task [15], [16], or the Switchboard task, where the adaptation algorithm proposed in this paper is particularly applicable because 1) the computational complexity is not a major constraint in these tasks, and 2) the amount of available adaptation data is very limited (15 sentences in the *WSJ* task, as few as one or two sentences in the Hub 4 task), and from the experimental results of Section IV-B, for small amounts of adaptation data, the algorithm presented in this paper can provide a fair amount of performance improvement over that provided by the MLLR technique.[6]

VI. Conclusion

A speaker-adaptation strategy was described that is based on finding a subset of training speakers in the training corpus who are acoustically the most similar to the test speaker, and then computing a set of linear transforms for each of the selected training speakers that maps the training speaker's data closer to the acoustic space of the test speaker. The Gaussians of the speaker-independent system are then reestimated using the transformed data from the selected training speakers. The scheme is computationally more complex than other adaptation schemes such as [5]; however, it is seen to provide a fair amount of gain over these other schemes. Comparisons with the performance of speaker-dependent systems (estimated from 1200 sentences) also showed that this adaptation scheme is able to go more than half the way to speaker-dependent performance with as little as three sentences of adaptation data. Also, the performance of the scheme does improve with the amount of adaptation data; however, this improvement is not very large, and most of the gain is obtained with the first few sentences of adaptation data. The main applicability of the adaptation scheme is felt to be in tasks such as the ARPA-sponsored *WSJ* and Hub 4 tasks, etc., where the amount of adaptation data is very limited, and computational complexity is not a major constraint.

Acknowledgment

The authors thank the reviewers and the associate editor, Dr. M. Rahim, for their detailed perusal of the manuscript and their comments.

References

[1] *Proc. ARPA Speech and Natural Language Workshop*, Jan. 1995. Austin, TX.
[2] L. Bahl, P. de Souza, P. Gopalakrishnan, and M. Picheny, "Context-dependent vector quantization for continuous speech recognition," in *Proc. ICASSP*, Minneapolis, MN, Apr. 1993, pp. II-632–635.

[6] Further, it is possible to simplify the algorithm by preclustering the training speakers into a small number of clusters (say, 8 or 16), and to make up a model for each cluster, instead of treating each training speaker individually. As the number of clusters is much less than the number of speakers, it would not be expensive to store the individual cluster models. The speaker selection process would now be replaced by a cluster selection process, with a separate set of transformations being made for each cluster, and the model parameters reestimated from the transformed models of the closest clusters. This is an area for future research.

[3] L. R. Bahl *et al.*, "Robust methods for using context-dependent features and models in a continuous speech recognizer," in *Proc. ICASSP*, Adelaide, Australia, May 1994, pp. I-533–536.
[4] L. R. Bahl *et al.*, "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal task," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 41–44.
[5] C. J. Legetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMM's," in *Comput. Speech Lang.*, vol. 9, pp. 171–186, 1996.
[6] V. Digalakis and L. Neumeyer, "Speaker adaptation using combined Transformation and Bayesian methods," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 680–683.
[7] J. R. Bellergarda *et al.*, "Experiments using data augmentation for speaker adaptation," in *Proc. ICASSP*, Detroit, MI, May 1995, pp. 692–695.
[8] J. L. Gauvain and C. H. Lee, "Maximum-*a-posteriori* estimation for multivariate Gaussian observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.
[9] G. Zavagliogkos, R. Schwartz, and J. Makhoul, "Batch, incremental and instantaneous adaptation techniques for speech recognition," in *Proc. ICASSP*, Detroit, MI, May 1995, pp. 676–679.
[10] M. Padmanabhan *et al.*, "Speaker clustering and transformation for speaker adaptation in large-vocabulary speech recognition systems," in *Proc. ICASSP*, Atlanta, GA, May 1996, vol. II, pp. 701–704.
[11] A. Sankar *et al.*, "Training data clustering for improved speech recognition," in *Proc. EUROSPEECH*, Madrid, Spain, Sept. 1995, pp. 503–506.
[12] T. Anastasakos *et al.*, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996, pp. 1137–1140.
[13] T. Kosaka and S. Sagayama, "Tree-structured speaker clustering for fast speaker adaptation," in *Proc. ICASSP*, Adelaide, Australia, May 1994, pp. I-245–I-248.
[14] T. Kosaka, J. Takami, and S. Sagayama, "Rapid speaker-adaptation using speaker-mixture allophone models applied to speaker-independent speech recognition," in *Proc. ICASSP*, Adelaide, Australia, May 1994, pp. II-570–II-573.
[15] P. S. Gopalakrishnan *et al.*, "Transcription of radio broadcast news with the IBM large vocabulary speech recognition system," in *Proc. Spoken Language Technology Workshop*, Feb. 1996, pp. 72–76.
[16] P. S. Gopalakrishnan *et al.*, "Acoustic models used in the IBM system for the ARPA hub 4 task," in *Proc. Spoken Language Technology Workshop*, Feb. 1996, pp. 77–80.

Mukund Padmanabhan (S'89–M'92) received the B.Tech. (Hons) degree from the Indian Institute of Technology, Kharagpur, India, in 1987, and the M.S. and Ph.D. degrees from the University of California, Los Angeles, in 1989 and 1992, respectively.

Since 1992, he has been with the Speech Recognition Group at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. His research interests are in speech recognition algorithms, signal processing algorithms, and analog integrated circuits.

Lalit R. Bahl (S'66–M'68–SM'94–F'96) received the Ph.D. degree in electrical engineering from the University of Illinois, Urbana-Champaign, in 1969.

He joined the Thomas J. Watson Research Center, Yorktown Heights, NY, in 1968. Until 1972, he worked on several research projects relating to communications. In 1972, he joined the center's Speech Recognition Group, where he is currently Manager of speech recognition algorithms.

David Nahamoo (S'78–M'81) received the B.S. degree from Tehran University, Iran, the M.S. degree from Imperial College of London, U.K., and the Ph.D. degree from Purdue University, West Lafayette, IN, all in electrical engineering, in 1975, 1976, and 1982, respectively. During these years, he worked on algorithms for x-ray tomography, ultrasonic diffraction and echo imaging.

Since 1982, he has been a member of the Continuous Speech Recognition Group, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he is currently Senior Manager of Human Language Technologies. He is head of research efforts in speech recognition and natural language understanding. He has worked in many areas of speech recognition, such as noise and environment robustness, robust training with small amounts of data, speaker mapping, HMM modeling, training, and decoding.

Michael A. Picheny (S'73–M'81) was born in New York, NY, on July 2, 1954. He received the S.B., S.M., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1975, 1978, and 1981, respectively. While at MIT, he worked on aids for people with hearing impairments.

Since 1981, he has been working on various problems in speech recognition at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. He is currently Manager of the Speech Recognition Technologies Group, where he focuses on algorithm development for the next generation of IBM speech recognition products. He is co-author of over 20 patents in the area of speech recognition.

Dr. Picheny was an Associate Editor for IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He has received several IBM awards.