A Novel Algorithm For Rapid Speaker Adaptation Based On Structural Maximum Likelihood Eigenspace Mapping

Bowen Zhou and John H. L. Hansen

Robust Speech Processing Laboratory, The Center for Spoken Language Research
University of Colorado at Boulder, Boulder, CO, 80302, U.S.A
{zhoub, jhlh}@cslr.colorado.edu   Web: http://cslr.colorado.edu

Abstract

In this paper, we propose a novel algorithm for rapid speaker adaptation based on our Structural Maximum Likelihood Eigenspace Mapping (SMLEM). The proposed method constructs a binary-tree structured hierarchical Speaker Independent (SI) eigenspace at different levels from well-trained SI system models, and then dynamically constructs a new set of speaker dependent (SD) eigenspaces at corresponding levels, according to the availability of incoming adaptation data. By mapping the mixture Gaussian components from a SI eigenspace to SD eigenspaces in a maximum likelihood manner, the SI models are adapted towards SD models (EM algorithm is used to derive the eigenspace bias). Compared with conventional MLLR, the proposed algorithm is both computationally cheaper and more effective when only a very small amount (from 5 to 15 seconds) of adaptation data is available. In our simulations using the DARPA WSJ Spoke3 corpus, an average of 10.5% relative reduction in WER was achieved over MLLR adaptation when using 5 seconds data for adaptation.

1. Introduction

Rapid speaker adaptation has been an interesting and challenging problem for Large Vocabulary Continuous Speech Recognition (LVCSR) for some time. The task of how to adapt a set of speaker independent (SI) models to a new speaker with a small amount of adaptation data is very important in many applications. Currently the most commonly-used speaker adaptation algorithms are MLLR [5] and MAP [4], as well as several variations of these two schemes (see reviews in [6]). These two families of algorithms can obtain direct adaptation for the test speaker with only transforming the SI models. Recently, a family of cluster-based speaker adaptation schemes have received much attention [2]. In this approach, the correlations among different training speakers are explored and adaptation is based on obtaining the appropriate linear combination of acoustic models of different training speakers, in terms of the distance of the test speaker to each training speaker. This family of schemes was shown to produce better speaker adaptation performance than MLLR or MAP when only a small amount of adaptation data was available. Eigenvoice, which is based on prior knowledge of speaker variation, is a typical example of cluster-based speaker adaptation [2]. In this method, the speaker space is constructed by spanning a K-space via Principal Component Analysis (PCA) of super vectors. Next, the target speaker is represented as a point in this K dimensional eigenspace. However, there are several obstacles for applying the Eigenvoice method in LVCSR tasks. First, Eigenvoice modeling does not obtain the adapted models from a single set of SI models, instead, it requires sufficient well-trained SD systems to construct the speaker space. Second, the PCA analysis is particularly difficult and numerical problems can result [6] for large scale HMM systems that usually contain more than 100K Gaussians. Similar analysis of other speaker cluster-based schemes indicate that they too require either the entire training corpus to be available on-line for the adaptation process, or a set of SD models. These issues impact the practical application of model adaptation due to either large data storage requirements or insufficient reliable data to obtain robust SD models. If this class of model adaptation methods are collectively compared, it becomes apparent that an algorithm that directly adapts acoustic models from a single set of SI models is more attractive. The goal therefore, is to develop an algorithm that can sufficiently capitalize on information contained in the SI model set, yet retains desirable adaptation performance when only small amounts of adaptation data are available.

In this paper, we formulate a novel algorithm for rapid speaker adaptation in LVCSR using Structural Maximum Likelihood Eigenspace Mapping (SMLEM). The proposed method constructs a binary structured hierarchical SI eigenspace at different levels from well-trained SI models, and dynamically constructs a corresponding set of speaker dependent eigenspaces according to the availability of incoming adaptation data for the test speaker. By mapping mixture Gaussians from the SI eigenspace to a SD eigenspace in a maximum likelihood manner (the Expectation-Maximum algorithm is used to derive the mapping position), the SI models are thereby adapted directly towards speaker dependent ones. The proposed method is able to adapt SI models effectively with a very small amount of adaptation data. Theoretically, SMLEM is much more efficient than cluster-based adaptation and computationally cheaper than MLLR. With a very limited amount of adaptation data, SMLEM outperforms MLLR since SMLEM uses a smart way to estimate the transformation matrix and hence only a d-dimension bias vector need be estimated, a stark contrast with the burden of estimating $d \times (d+1)$ parameters for MLLR adaptation.

This paper is organized as follows: Sec. 2 describes the SMLEM algorithm in a step by step manner; Sec. 3 evaluates the algorithm using data from the DARPA WSJ Spoke3 corpus and compares it with MLLR. Sec. 4 is the discussion and Sec. 5 summarizes our contributions.

This work was supported by NSF under Cooperative Agreement No. IIS-9817485. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

2. SMLEM: Structural Maximum Likelihood Eigenspace Mapping

2.1. Motivation

In speech recognition, raw speech from a speaker is typically first converted into the cepstrum. One interesting observation is that the covariance matrix of the cepstrum of that speaker reflects a range of speaker dependent features. An example can be found in [7] where the statistics based on covariances are used successfully to detect speaker turns in audio streams. The eigenvectors of such covariance matrices, which are positive definite, will construct the eigenspace for that speaker. Our initial motivation is to adapt the SI models to SD ones by mapping the SI component Gaussians according to the eigendirections in both SI space and test speaker's space. In more detail, we can construct a SI eigenspace by first computing the covariance matrix of the SI component Gaussian means and then extracting the eigenvectors of such a covariance matrix to construct an eigenspace. In this eigenspace, each Gaussian mean is represented by a point and is distinct from others by occupying different eigen-positions in the space (i.e, the projections of this Gaussian to eigenvectors is decided by phoneme acoustics dependent on human speech production). From the test speaker side, we can perform a similar analysis to construct a test speaker specified eigenspace from adaptation frames. Hence, the Gaussian adaptation problem can be viewed as a task of how to map an original component Gaussian μ from the SI eigenspace to $\hat{\mu}$ of the SD eigenspace. One simple mapping is to assume that these two Gaussians share the same first principal component [3] in their associated eigenspaces, which captures the new speaker's most distinguished variations in his Gaussian space. On the other hand, the SI and adapted Gaussians should be somewhat acoustically close from the viewpoint of phoneme acoustics. Therefore, let $E_{si} = [e^1_{si}, e^2_{si}, \ldots, e^d_{si}]$ represent the SI eigenspace, where e^i_{si} is the normalized ith eigenvector associated with the ith largest eigenvalue of the covariance of SI component Gaussians and d is the eigen-dimension, next let $E_{test} = [e^1_{test}, e^2_{test}, \ldots, e^d_{test}]$ denote the eigenspace estimated from the test speaker's adaptation data. With this representation, we can roughly estimate each speaker dependent Gaussian mean $\hat{\mu}$ by:

$$E_{test} \times \hat{\mu} = E_{mix} \times \mu \tag{1}$$

where $E_{mix} = [e^1_{si}, e^2_{test}, \ldots, e^d_{test}]$, noting that the first eigen eigenvector is obtained from SI eigenspace. It is clear from Eq. 1 that the adapted mean is the projection of original SI mean to the subplane that shares the same first principal component in test speaker's eigenspace with that of the SI mean in the SI eigenspace (i.e., the adapted mean is the closest neighbor of the SI mean that keeps the first principal component unchanged in test speaker's eigenspace). In this way, we capture the largest variance of the test speaker and hence make the adapted mean discriminative for the test speaker. For convenience, we can rewrite Eq. 1 as:

$$\hat{\mu} = {E_{test}}^{-1} \times E_{mix} \times \mu \tag{2}$$

where ${E_{test}}^{-1} \times E_{mix}$ gives the associated eigenspace transformation for each Gaussian mean vector.

2.2. Structural Eigenspace Mapping

The method described above is only a rough adaptation scheme since it constructs only a single space for both the SI models and the test speaker. The space is constructed from so many broad Gaussians that it is difficult to distinguish a specific Gaussian's principal components from others. The adaptation performance will hence saturate soon as more adaptation data becomes available. As an alternative, partitioning the Gaussian means or adaptation frames into too many clusters will cause poor eigenspace estimation when there is only limited adaptation data available. Therefore, we propose a structural eigenspace mapping scheme to allow hierarchical mapping according to the amount of available adaptation data. In this method, all of the component Gaussian means of the well-trained SI system are clustered into N base classes in a top-down splitting manner using the K-means algorithm according to their acoustic similarity. Next, a binary tree, which will ultimately contain $2N - 1$ tree nodes, is generated in a bottom-up manner from these base classes using a Euclidean distance measure. In the adaptation stage, we first estimate the mean and *full* covariance matrix for both test speaker and SI sides at selected nodes in this tree. There are several options to achieve this goal.

The first method is to estimate these statistics for the test speaker directly in a maximum likelihood manner through a modified version of the Baum-Welch algorithm. The mean $\hat{\mu}^{(n)}_{test}$ and full covariance matrix $\hat{\Sigma}^{(n)}_{test}$ of the n-th base class $C^{(n)}$ in the eigenspace tree are found as follows:

$$\hat{\mu}^{(n)}_{test} = \frac{\sum_{t=1}^{T} \sum_{C^{(n)}} \gamma^{(s)}_m(t) O(t)}{\sum_{t=1}^{T} \sum_{C^{(n)}} \gamma^{(s)}_m(t)} \tag{3}$$

and

$$\hat{\Sigma}^{(n)}_{test} = \frac{\sum_{t=1}^{T} \sum_{C^{(n)}} \gamma^{(s)}_m(t)(O(t) - \hat{\mu}^{(n)}_{test})(O(t) - \hat{\mu}^{(n)}_{test})^T}{\sum_{t=1}^{T} \sum_{C^{(n)}} \gamma^{(s)}_m(t)} \tag{4}$$

Note that the summation in the above equations are conducted for all Gaussians that belong to the n-th base class in the tree.

The second method is to consider only those component Gaussians that have sufficient observations in the adaptation data (i.e., whose $\sum_{t=1}^{T} \gamma^{(s)}_m(t)$ is bigger than some threshold). The Maximum Likelihood Estimation (MLE) of these Gaussians are computed first, and the class covariance can be found as follows:

$$\hat{\Sigma}^{(n)}_{test} = \frac{1}{T_n} \sum_{\substack{C^{(n)} \\ obs.}} (\hat{\mu}^{(s)}_m - \hat{\mu}^{(n)}_{test})(\hat{\mu}^{(s)}_m - \hat{\mu}^{(n)}_{test})^T \tag{5}$$

where $\hat{\mu}^{(s)}_m$ is the MLE of the Gaussian mean for mixture m in state s for test speaker, T_n is the number of sufficiently observed Gaussians in the adaptation data for the n-th class.

In both cases, the accumulation of higher level nodes in the tree are obtained by summing the accumulations of their child nodes, and the node mean and covariance matrix from the SI side are estimated by:

$$\hat{\mu}^{(n)}_{si} = \frac{1}{T_n} \sum_{\substack{C^{(n)} \\ obs.}} \mu^{(s)}_m \tag{6}$$

and

$$\hat{\Sigma}^{(n)}_{si} = \frac{1}{T_n} \sum_{\substack{C^{(n)} \\ obs.}} (\mu^{(s)}_m - \hat{\mu}^{(n)}_{si})(\mu^{(s)}_m - \hat{\mu}^{(n)}_{si})^T \tag{7}$$

where T_n is the number of Gaussians contributing to the n-th class covariance estimation for the given test speaker. In our

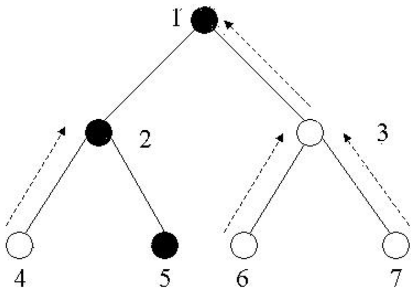

Figure 1: *A portion of the binary tree structure of hierarchical eigenspaces.*

experiments, the second method shows better performance and hence we will use this one in our evaluation.

When adaptation data comes in, the level of the eigenspace mapping will be decided based on the amount of adaptation data. Using a bottom-up scheme to traverse the binary tree, if the T_n in Eq. 5 is greater than some established threshold, then the eigenspace mapping will be conducted at this level. Alternatively, we will continue to traverse that tree until the root is reached. Fig. 1 gives an example of how a portion of the tree is traversed. In this figure, the dark nodes are those that have sufficient adaptation data while white nodes have insufficient data. The arrows show how lower level nodes map to higher level nodes to accumulate data.

2.3. Maximum Likelihood Estimation of Eigenspace Bias

In practice, the assumption that Gaussians share the same principal components in the SI and test speaker-dependent eigenspaces is not necessary true, especially for non-native speakers (i.e., variations caused by accent and dialect). Keeping this fact in mind, we modify our eigenspace mapping equation by allowing an eigenspace bias to exist for the test speaker. For the mth Gaussian mixture of state s that belongs to the nth eigenspace, the eigenspace mapping equation is changed to:

$$E_{test}^n \times \hat{\mu}_m^{(s)} = E_{mix}^n \times \mu_m^{(s)} + \hat{b}^n, \tag{8}$$

where $\hat{b}^n$ is the bias in the test speaker's nth eigenspace. We can see later that this bias can be estimated at more concrete levels. We would like to derive $\hat{b}_n$ in a manner that maximizes the adaptation data likelihood $P(O|\lambda)$ given model λ. According to the EM algorithm [1], we define an auxiliary function $Q(\lambda, \hat{\lambda})$:

$$Q(\lambda, \hat{\lambda}) = -\frac{1}{2}\sum_t\sum_s\sum_m \gamma_m^{(s)}(t)\log f(O(t)|\hat{\lambda}) \tag{9}$$

Since we only adapt Gaussian means, we can simply write:

$$\log f(O(t)|\hat{\lambda}) = (O(t) - \hat{\mu}_m^{(s)})^T {\Sigma_m^{(s)}}^{-1}(O(t) - \hat{\mu}_m^{(s)}) \tag{10}$$

To maximize $Q(\lambda, \hat{\lambda})$, we set $\frac{\partial Q(\lambda,\hat{\lambda})}{\partial b} = 0$ and are therefore able to derive the following equation:

$$\sum_t\sum_s\sum_m \gamma_m^{(s)}(t){\Sigma_m^{(s)}}^{-1}(O(t) - {E_{test}^n}^{-1}E_{mix}^n\mu_m^{(s)}) = \sum_t\sum_s\sum_m \gamma_m^{(s)}(t){\Sigma_m^{(s)}}^{-1}{E_{test}^n}^{-1}\hat{b}^n \tag{11}$$

The accumulation of each side of Eq. 11 is conducted first for each base class. Next, according to the hierarchical eigenspace structure, the base classes that belong to the same eigenspace are summed for high level accumulations until the root of the tree is reached. The choice of tree levels on which to estimate the eigenspace bias is very flexible, and it is not necessary to be the same as that used for eigenspace transformation estimation. Instead, we perform a bottom-up traversing of the binary tree and stop at the lowest nodes whose $\gamma_n = \sum_{t=1}^{T}\sum_{C(n)}\gamma_m^{(s)}(t)$ is bigger than some threshold.

From Eq. 11 we can see that we only have a d-dimensional vector to solve and one equation to accumulate, and therefore the computational cost is much less expensive than the $d \times (d+1)$ parameter estimation needed by MLLR.

3. Evaluations

The proposed algorithm was evaluated using the 1993 DARPA WSJ Spoke3 task corpus. The Spoke3 data consists of 10 non-native speakers of American English. Each speaker provided 40 utterances for model adaptation and another 40 utterances for testing. We selected the last 6 speakers from this corpus for our adaptation experiments. The first 20 test utterances from each test speaker are chosen for the recognition test. For adaptation, the first N adaptation utterances of each speaker, which are different from the test utterances, are selected. Here we are mainly interested in rapid speaker adaptation with a small amount of adaptation data, and hence we consider $N = 1$ and $N = 3$ adaptation utterances in our evaluation, which roughly contains 5 and 15 seconds of adaptation speech for each speaker, respectively. The baseline speech recognition system we used is the CMU SPHINX-3 LVCSR. The acoustic models are based on continuous HMMs with 100K component Gaussians (6275 senones each of which has 16 mixtures), a 39 dimensional feature vector consisting of MFCC cepstra, delta cepstra and double delta cepstra. The language model is a 5000-word trigram. Our adaptation experiments are conducted in a supervised manner. We compare our SMLEM algorithm with the standard MLLR scheme (note that since we only have limited adaptation data, one global class is used in MLLR adaptation).

In our SMLEM implementation, the 100K Gaussians in the SI models are clustered into 128 base classes and a binary tree was generated from these base classes. The static and dynamic parts of the feature vectors are considered as different streams. The overall adaptation process including class covariance estimation, PCA analysis, eigenspace mapping and bias estimation are all conducted independently for different streams. In our current implementation and evaluation, however, we only adapt the static parts of the feature vectors and keep dynamic parts unchanged from the SI models. The energy and its dynamic parts are kept unchanged as well. The $\gamma_m^{(s)}$ threshold for defining sufficient observed Gaussian in adaptation data is 1.0. The minimum requirement for a node in the tree to conduct PCA analysis is to contain 30 or more sufficiently observed component Gaussians. The γ_n threshold level to estimate eigen-bias for a static stream is set to 10.0.

Table 1 compares the WERs of MLLR and SMLEM using only ONE adaptation utterance with that from the baseline performance. The last column shows the relative gain of SMLEM over MLLR. We can see that MLLR provides reasonable improvement for only two of the six speakers, with an average absolute 0.3% *increase* in WER when compared to the baseline recognizer. This is to be expected since only a very limited amount of adaptation data is available. On the other hand, the SMLEM adaptation scheme provides consistent improvements over both baseline and MLLR adaptation for every speaker. The SMLEM reduces the baseline WER by absolute 1.5%, and the relative WER reductions of SMLEM over MLLR for different speakers are ranged from 1.3% to 22.7% with an average of 10.5%. We also point out an important factor in that for overall SMLEM adaptation. For our SMLEM scheme, the computational speed is more than 10 times faster than MLLR. This would allow SMLEM adaptation to be applied more often than MLLR.

Speaker	BASELINE	MLLR	SMLEM	Rel. Impr.
4n5	25.5	23.2	22.9	1.3%
4n8	17.9	19.6	16.8	16.7%
4n9	10.4	11.9	9.7	22.7%
4na	13.5	13.2	12.9	2.3%
4nb	28.4	32.1	28.1	14.3%
4nc	13.9	11.2	9.2	21.7%
Avg.	18.7	19.0	17.2	10.5%

Table 1: *Evaluation (% WER) of Speaker Adaptation Algorithms using only ONE adaptation utterance (5 sec.)*

Next, the results in Table 2 compare WER using three adaptation utterances for MLLR and SMLEM. It is clear that SMLEM also outperforms MLLR with approximately 15 seconds of adaptation data. For SMLEM, some speakers showed measurable improvement over the one utterance adaptation test (e.g, speaker 4n5, 4nb); while others showed modest levels of improvement (e.g, 4n9, 4nc). The overall average WER reduction over baseline continues to improve from the absolute 1.5% decrease for one adaptation utterance to absolute 2.5% with 3 adaptation utterances.

Speaker	BASELINE	MLLR	SMLEM	Rel. Impr.
4n5	25.5	21.4	20.8	2.8%
4n8	17.9	19.6	18.2	7.5%
4n9	10.4	9.3	9.7	-4.1%
4na	13.5	12.9	13.5	-4.4%
4nb	28.4	24.1	22.9	5.5%
4nc	13.9	10.5	9.9	6.0%
Avg.	18.7	16.6	16.2	2.5%

Table 2: *Evaluation (% WER) of the Speaker Adaptation Algorithms using three adaptation utterances (15 sec.)*

4. Discussion

The proposed SMLEM algorithm is similar in its goal with conventional MLLR. Both methods adapt SI means by a linear transformation and then follow by a shift. However, MLLR jointly estimates the transformation and the shift as a $d \times (d+1)$ matrix in a maximum likelihood manner while SMLEM does this in a pipeline manner. First, the $d \times d$ transformation matrix is estimated in a smart way via eigenspace mapping; then the d-dimensional shift is estimated in eigenspace to maximize the adaptation data likelihood. We can see that SMLEM only has a d-dimensional vector to be estimated, which makes it possible to obtain robust results when only a small amount of adaptation data is available. This makes the adaptation much faster, and therefore allows for more frequent applications of model adaptation given computational and data resources. Compared to other adaptation schemes who also achieve improved performance for limited amounts of adaptation data (e.g, Eigenvoice etc), SMLEM does not require many speaker dependent models or an online training corpus. SMLEM adapts HMMs for a test speaker directly from the SI models, and hence is more efficient to implement. Although both SMLEM and Eigenvoice use PCA, the former does it in feature spaces while the latter is in a speaker space.

While SMLEM model adaptation has been shown to be effective for the DARPA WSJ Spoke3 task, there are several issues presently under investigation. For example, how does the performance change as more adaptation data becomes available? Should the algorithm settings which showed consistent improvement for short adaptation data be adjusted as more adaptation data comes in? At this time, it is apparent that SMLEM adapted means are a better replacement for the prior distribution in MAP estimation than using SI means. It would therefore be logical to combine both SMLEM and MAP in the future. The performance improvement in our evaluation are obtained from adapting the static parts of the feature vectors and no gains have been observed for adapting dynamic features. How to adapt dynamic features within the framework of SMLEM? Finally, an issue of how to best estimate the principal components from the adaptation data for both test speaker and SI models is also worthy of further research.

5. Conclusions

In this paper, we have formulated a novel algorithm for rapid speaker adaptation in LVCSR using our proposed Structural Maximum Likelihood Eigenspace Mapping (SMLEM) method. lThe proposed method has been shown to adapt SI models more effectively than MLLR (results show an average 10.5% relative improvement), in addition to being approximately 10 times faster than MLLR, with very small amounts of adaptation data.

6. References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood Estimation from Incomplete Data Via the EM algorithm”, J. Roy. Statist., Soc.B39, pp. 1-38

[2] R. Kuhn, et al, “EigenVoices for Speaker Adaptation”, *Proc. ICSLP’98,* Sydney, Australia, Nov., 1998

[3] N. C. Giri, “Multivariate Statistical Analysis,” *Chap. 10, Marcel Dekker, Inc,* Nov., 1995

[4] J.-L. Gauvain and C.-H. Lee, “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,” *IEEE Trans. Speech Audio Proc.,* Vol. 2, pp. 291-298, Apr. 1994

[5] C. Leggetter and P. Woodland, “Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models,” *Comp.Speech Lang.,* Vol. 9 pp.171-185, 1995

[6] P. C. Woodland, “Speaker Adaptation: Techniques and Challenges,” *IEEE Workshop on Auto. Speech Recog. & Under.,* pp.85-90, Colorado, 1999

[7] B. Zhou and J. H. L. Hansen, “Unsupervised Audio Stream Segmentation and Clustering Via the Bayesian Information Criterion,” *Proc. ICSLP’2000,* pp. 714-717, Beijing, China, 2000