ADAPTATION EXPERIMENTS ON THE SPINE DATABASE WITH THE EXTENDED MAXIMUM LIKELIHOOD LINEAR TRANSFORMATION (EMLLT) MODEL

Ramesh Gopinath, Vaibhava Goel, Karthik Visweswariah & Peder Olsen

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
{rameshg,vgoel,kv1,pederao}@us.ibm.com

ABSTRACT

This paper applies the recently proposed Extended Maximum Likelihood Linear Transformation (EMLLT) model for inverse covariances in a Speaker Adaptive Training (SAT) context. The paper adapts standard algorithms for maximum likelihood estimation of linear transforms for mean, variance and feature space adaptation respectively, to the EMLLT model. Experimental results showing word-error-rate improvements are reported on the SPINE2 database. The system described here is the best-performing system submitted by IBM in the SPINE2 evaluation conducted by NIST in October 2001.

1. INTRODUCTION

Recently the EMLLT model ([1, 2]) was proposed as an alternative to the standard diagonal covariance Gaussian Mixture Models used to model the HMM states in state-of-the-art speech recognition systems. The basic idea in EMLLT is to model the inverse covariance (precision) matrix of all the Gaussians in a basis of appropriate dimension. By choice of this dimension/basis the EMLLT allows one to gradually vary the complexity of the model from diagonal covariances on the one extreme to full-covariances in the other. On several databases it has been shown recently that the EMLLT model can give significant gains in performance. The experiments with EMLLT in [1] were conducted in the context of standard MFCC and MFCC+LDA-based features. This paper investigates the use of the EMLLT model in a Speaker Adaptive Training (SAT) setting where the EMLLT model is built on a canonical feature space [3, 4]. Each training speaker's data is first Vocal-Tract-Length (VTL) normalized and then subjected to a speaker-specific affine transformation. The goal of these two transformations is to reduce as much as possible the speaker-specific features in the training data. A standard Gaussian mixture model is built for the HMM states in this canonical feature space. This paper studies the effect of an EMLLT model built in this canonical feature space. To apply such a model at test time it is necessary to first VTL normalize the test speaker's data and then estimate a maximum likelihood affine feature space transformation. Further improvements in accuracy can be obtained by adapting the means and precision matrices of the Gaussians using linear transformations. The well-known estimation of the maximum likelihood (ML) linear adaptation transformation transformations in the diagonal covariance model case can be adapted to the EMLLT case.

2. EMLLT MODEL

In the EMLLT model the d-dimensional acoustic features (be it original MFCC or canonical VTL+SAT features) are modeled with Gaussians of a special form. Specifically the precision matrix of the j^{th} Gaussian, say $P_j (= \Sigma_j^{-1})$ is of the form

$$C\Lambda_j C^T, \quad C \in \mathbf{R}^{d \times D}, \Lambda_j \in \mathbf{R}^{D \times D}, \tag{1}$$

Λ_j diagonal and $d \leq D \leq d(d+1)/2$. The parameters of this model are C, Λ_j (and of course the means m_j and priors π_j). These parameters can be estimated in an ML fashion using a generalized EM algorithm [1].

3. ADAPTATION OF EMLLT

In the speech recognition literature three forms of adaptation with linear transformations are popular - adaptation of means (MLLR), adaptation of precisions (also known as Full-Variance Transform) and adaptation of features (also known as FMLLR and Constrained MLLR) [3]. We adapt these to the EMLLT model. Given some test data, let $\gamma_j(t)$ denotes the posterior probability of Gaussian j for speech frame x_t (from the E-step). Then, the sufficient statistics for estimating the parameters of all three forms of adaptation in the M-step are given by

$$\begin{aligned}
G_1^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk} x_t x_t^T, \qquad (2)\\
G_2^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk} x_t,\\
G_3^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk} x_t \mu_j^T,\\
G_4^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk} \mu_j \mu_j^T,\\
G_5^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk} \mu_j,\\
G_6^{(k)} &= \sum_{j,t}^{J,T} \gamma_j(t) \Lambda_{jkk},\\
\beta &= \sum_{j,t}^{J,T} \gamma_j(t).
\end{aligned}$$

0-7803-7402-9/02/$17.00 ©2002 IEEE

The advantage of computing and storing the adaptation statistics in the form above is that all three forms of adaptation can be applied in any order and/or iteratively with the same statistics.

3.1. Adaptation of Means

Since $\hat{\mu}_j^s = A^s \mu_j + b^s$, the Q function, say $Q_{\rm MT}$ is,

$$-1/2 \sum_{j,t} \gamma_j(t)(x_t - A\mu_j - b)^T C \Lambda_j C^T (x_t - A\mu_j - b).$$

$Q_{\rm MT}(A, b)$ is quadratic and hence strictly concave in the parameters (A, b). The difference between this case and standard MLLR is that the problem no longer breaks down into independent subproblems when one considers one row at a time. For large d (say $d = 60$ as in one example in this paper) directly solving this quadratic in $d^2 + d$ variables is very slow. We propose using a conjugate-gradient algorithm with pre-conditioning to solve this optimization problem [5]. The pre-conditioner used is the matrix of diagonal elements of the positive-definite matrix associated with the quadratic form $Q_{\rm MT}$. The cost of the algorithm is roughly the cost of computing the gradient once per iteration. A formula for the gradient of $Q_{\rm MT}$ explicitly in terms of the statistics in equation (3) is given at the end of the paper.

3.2. Adaptation of Features

Since we transform the feature vectors as $\hat{x}^s = A^s x + b^s$ the Q function, say $Q_{\rm FT}(A, b)$, is

$$\begin{aligned} Q_{\rm FT} &= -\sum_{j,t} \gamma_j(t)(Ax_t + b - \mu_j)^T C \Lambda_j C^T (Ax_t + b - \mu_j) \\ &\quad +2\beta \log|\det A|. \end{aligned}$$

Because of the additional $\log|\det A|$ term Q_{FT} is not concave in (A, b). However, considered as a function of one row of A (with all other parameters fixed) and searching only in the region where $\det A$ has the same sign as the initial point Q_{MT} is a strictly concave function with a unique maximizer. This is because the determinant of a matrix is linear in a row of that matrix given that all the other rows are fixed. Setting the gradient with respect to a_j the $j^{\rm th}$ row of A, equal to zero we get

$$\beta \frac{v_j}{v_j^T a_j} = R_j a_j + s_j, \tag{3}$$

where v_j is a cofactor vector satisfying $v_j^T a_j = \det A$ and R_j and s_j are matrices that depend on the sufficient statistics in $G_1^{(k)}, \ldots, G_6^{(k)}$ in equation (3). The exact dependence of R_j and s_j on the statistics is given at the end of the paper. Equation (3) is solved following a technique described in [3]. Basically we cycle through the rows of A, then optimize for b and repeat these two steps until convergence is achieved. The optimization in b is quadratic (see the end of the paper).

3.3. Adaptation of Precisions

The precision matrix is transformed as $\hat{P}_j{}^s = A^s P_j (A^s)^T$. Therefore the Q function, say Q_{PT} for this problem is

$$\begin{aligned} Q_{\rm PT} &= -\sum_{j,t} \gamma_j(t)(x_t - \mu_j)^T A C \Lambda_j C^T A^T (x_t - \mu_j) \\ &\quad +2\beta \log|\det A|. \end{aligned}$$

Formally this optimization problem is equivalent to the optimization problem in the case of feature space transforms. Hence one can adopt a similar row-by-row approach to optimize A as in the optimization of $Q_{\rm FT}$. However, for the precision transforms, we found it expedient to explicitly compute the function and the gradient and use conjugate gradient optimization function available in a numerical package. Details on the function and gradient.

3.4. Multiple Class-Dependent Transformations

The three types of transformations can be applied in any order iteratively to better match the EMLLT model to a test speaker's data. Notice that this does not require further recognition passes to get improved scripts. Also it is straightforward to apply these transformations in a class-dependent fashion by appropriately accumulating class dependent statistics in equation (3).

4. EXPERIMENTS ON SPINE

The SPINE database has speech collected from pairs of speakers who are engaged in a collaborative war-game. The database is split into four parts - the SPINE-1 training set (S1Tr), the SPINE-2 training set (S2Tr), the SPINE-1 evaluation set (S1E) and the SPINE-2 dry run set (S2D). There are 68 speakers in the entire database and since some of the speakers in S1Tr appear in S1E, a pruned set S1Ep (excluding these speakers) is used in our experiments. The actual SPINE task also requires automatic segmentation of the conversation. However, in this paper all experiments use a hand-segmented version of the SPINE database.

4.1. Baseline System

The IBM system for the SPINE evaluation ([6]) exploits combination of hypotheses from multiple recognizers [7]. As such two of the three baseline systems in [6] were used to SAT-train EMLLT acoustic models. The main difference between these two systems, say oRCC-16 and oPLP-16 respectively is the feature space in which the models are built. The feature space in oRCC-16 is 60 dimensional and is based on a variant of popular MFCC+LDA+MLLT with root-compression (instead of log-compression). The feature space in oPLP-16 is 39 dimensional and based on PLP cepstra followed by LDA+MLLT. In both oRCC-16 and oPLP-16 the final acoustic model is built in a canonical feature space, viz., a speaker dependent linear feature space transformation is applied to each training speaker's data. Preliminary experiments with EMLLT convinced us that the baseline SAT AMs had too many HMM states and too many Gaussians - EMLLT, an approximation to full-covariance Gaussians, was over-training. Therefore we fixed the training data alignments at the sub-phonetic level and built a collection of AMs varying the number of HMM states (leaves in decision-tree) and number of mixture components at each state. To ensure that any of these models could be used to initialize an EMLLT model (without over-training problems) care was taken so that each HMM state had at least 1000 training vectors and furthermore the number of mixture components was chosen using Bayesian Information Criterion [8]. Every acoustic model in the resulting collection had the property that no Gaussian was data-starved. The best performing model set in each of the two collections, RCC-16 and PLP-16, was used as a modified baseline for further EMLLT experiments. These modified baseline acoustic models, trained with fixed alignments and fewer HMM states and

mixture components, performed significantly better (see Table 1; the RCC-16 and PLP-16 numbers reported in [6] use this modified baseline).

4.2. SAT EMLLT Models

Ideally each training speaker's SAT transform, (A^s, b^s), should be estimated based on an EMLLT model. However, to save on computation the training SAT transforms were computed with the RCC-16 and PLP-16 AMs. Furthermore, the EMLLT models were built with fixed HMM-state level alignments (exactly the same as what was used in RCC-16 and PLP-16). The C-matrix (see (1)) in EMLLT was initialized as follows: the HMM states are split into groups based on acoustic-phonetic knowledge [1], and the MLLT matrices generated for each group are stacked on top of each other to give C. C is never changed for two reasons: a) concern of overtraining - SPINE has very little data and b) on two other tasks it has been observed that good performance can be obtained with this strategy [1]. Λ_j is updated using the generalized EM algorithm in [1].

4.3. Results and Analysis

Table 1 compares the performance of the SAT-trained EMLLT model corresponding to RCC-16 on the pruned Spine-1 evaluation test set (S1EP). The language model used in this experiment is described in [6] - essentially a trigram class-based LM trained on the SPINE acoustic training corpus. The EMLLT model had 120 basis elements (i.e., C is a 60×120 matrix) corresponding to MLLT directions of two groups of HMM states, viz., vowels and consonants. In all cases scripts from an initial-pass decode of the test data using a speaker-independent VTL-normalized acoustic model is used to compute the the statistics (3) and the speaker-specific adaptation transforms. The SAT-trained EMLLT model

	Model	% Word-Error-Rate
a	SAT oRCC-16	17.1
b	RCC-16	16.9
c	SAT-EMLLT	15.7
d	(c)+ 1 Mean Transform	15.1
e	(d) + 1 Precision Transform	15.3

Table 1. Word Error Rate Comparison of SAT Models on Spine-1 Pruned Evaluation Set (S1Ep): a) Baseline SAT oRCC-16 b) Modified SAT baseline RCC-16 c) SAT-trained 2d-EMLLT d) (c)+1 Mean Transform e) (d)+1 Precision Transform

clearly continues to give performance gain despite the fact that the baseline model is SAT-trained (which in turn, was clearly better than a speaker-independent or VTL-normalized model model). Further passes of adaptation give modest improvements. Since all the experiments used a fixed script for adaptation some amount of overtraining happens when one applies an MT and a PT on top of the FT (part of SAT).

Table 2 shows further results when using multiple feature space transforms on the S2D test set. Consistent gains are seen on SAT-trained EMLLT systems based on both the RCC-16 and PLP-16 baseline SAT models. Compared to typical gains using EMLLT on a non-SAT baseline (see [2] where gains up to 35% relative were reported) the SAT-EMLLT gains here are modest. To understand this better we plotted the density along a random projection for a

		RCC-16	PLP-16
a	Modified Baseline	22.9	25.1
b	(a)+multiple SAT Transforms	21.7	23.4
c	SAT-EMLLT	21.5	23.1
d	(c)+multiple SAT Transforms	21.0	22.9
e	(c)+1 Mean Transform	20.4	22.1

Table 2. Word Error Rate Comparison of SAT Models on Spine-2 Dry Run Test Set (S2D): a) Modified SAT Baseline b) (a)+multiple feature space transforms c) SAT-trained EMLLT d) (c)+Multiple Feature Transforms e) (c)+1 Mean Transform

random HMM state. From Fig. 4.3 it can be seen that the baseline SAT model's density *does* match the histogram quite well. The EMLLT model's density estimate is marginally closer to the histogram. This is in sharp contrast to similar plots on other databases (see [2]) where the baseline models poorly fit the histogram while the EMLLT model fits the histogram much better. Results on the SPINE database seem consistent with this observation.

5. APPENDIX

In this section we write down the functions that we need to optimize for adaptation in terms of the statistics $G_1^{(k)}, \ldots, G_6^{(k)}$.

5.1. Mean transform

$Q_{\mathrm{MT}}(A, b)$ for the mean transform is

$$-\sum_k c_k^T A G_4^{(k)} A^T c_k + 2\sum_k c_k^T A G_3^{(k)T} c_k$$
$$-2\sum_k c_k^T b G_5^{(k)T} A^T c_k - \sum_k b^T c_k c_k^T G_6^{(k)} b$$
$$+2\sum_k G_2^{(k)T} c_k c_k^T b.$$

A can be written as $A_0(i,j) + A_{ij} e_i e_j^T$, thus the partial derivative of $Q_{\mathrm{MT}}(A, b)$ with respect to the $(i,j)^{\mathrm{th}}$ element of A is

$$-A_{ij}\sum_k (c_k^T e_i)^2 (G_4^{(k)})_{jj} - \sum_k (c_k^T b)(G_5^{(k)^T} e_j c_k^T e_i)$$
$$+\sum_k c_k^T e_i e_j^T G_3^{(k)T} c_k - \sum_k c_k^T e_i e_j^T G_4^{(k)T} A_0^T c_k.$$

Also, the partial derivative of $Q_{\mathrm{MT}}(A, b)$ with respect to b is

$$-\sum_k G_6^{(k)} c_k c_k^T b - \sum_k (G_5^{(k)T} A^T c_k) c_k + \sum_k (c_k^T G_2^{(k)}) c_k$$

5.2. Feature transform

For feature space transforms we proceeded by optimizing the Q_{FT} with respect to each row in turn. The gradient of the Q_{FT} with respect to the j^{th} row of A is:

$$\beta \frac{v_j}{v_j^T a_j} - \sum_l \sum_k c_k^T e_j c_k^T e_l G_1^{(k)} a_l +$$
$$\sum_k c_k^T e_j (G_3^{(k)} - G_2^{(k)} b^T) c_k,$$

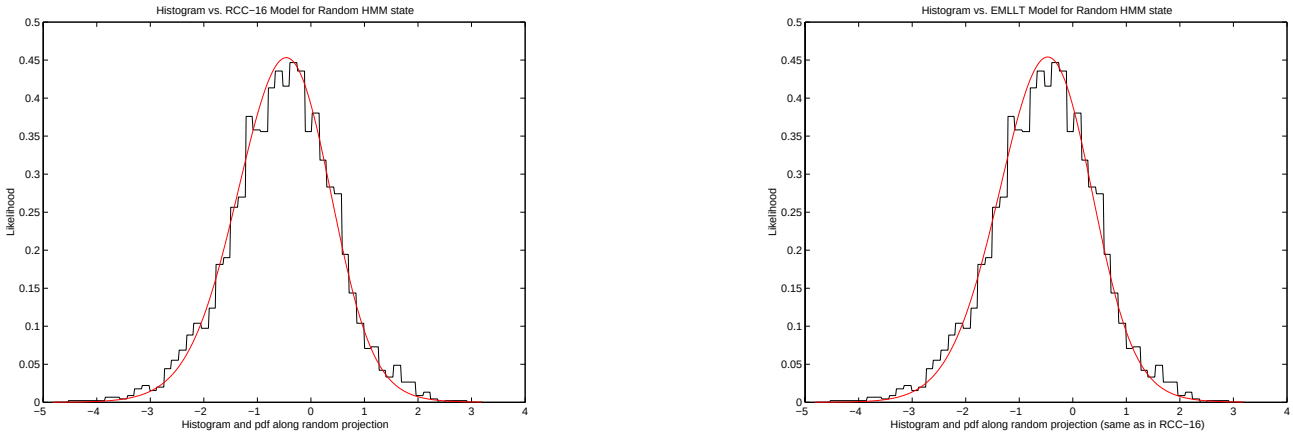

Fig. 1. Histogram vs. Model Density for Random HMM state along random projection

where v_j is a cofactor vector satisfying $v_j^T a_j = \det A$. We can set the gradient to zero and solve using the technique in [3]. As a function of b, $Q_{\mathrm{FT}}(A, b)$ is quadratic with the following gradient:

$$-\sum_k G_6^{(k)} c_k c_k^T b + \sum_k c_k (G_5^{(k)T} c_k - G_2^{(k)T} A^T c_k).$$

5.3. Precision transform

For precision transforms a numerical package implementing a conjugate gradient algorithm was used. It suffices to provide the computation of $Q_{\mathrm{PT}}(A)$ and its gradient to the optimization routine. Let $G^{(k)} = G_1^{(k)} + G_4^{(k)} - G_3^{(k)} - G_3^{(k)T}$, then $Q_{\mathrm{PT}}(A)$ is

$$\beta log|\det A| - \sum_k c_k^T A^T G^{(k)} A c_k.$$

The parial derivative of $Q_{\mathrm{PT}}(A)$ with respect to the the (i, j) element of A is:

$$\beta \frac{v_{ij}}{v_i^T a_i} - A_{ij} \sum_k (c_k^T e_j)^2 G_{ii}^{(k)} - \sum_k c_k^T e_j e_i^T G^{(k)} A_0^T c_k,$$

where A_0 is such that $A = A_0 + A_{ij} e_i e_j^T$ and v_i is a cofactor vector satisfying $v_i^T a_i = \det A$.

Acknowledgement

The authors would like to thank George Saon, Brian Kingsbury, Lidia Mangu and Mukund Padmanabhan for the initial acoustic models and help with the SPINE database.

6. REFERENCES

[1] P. A. Olsen and R. A. Gopinath, "Extended mllt for gaussian mixture models," *Transactions in Speech and Audio Processing*, 2001, submitted,http://www.research.ibm.com/people/r/rameshg/olsen-trans-sap2001.ps.

[2] P. Olsen and R. A. Gopinath, "Modeling inverse covariance matrices by basis expansion," in *ICASSP*, Orlando, Florida, 2002, submitted.

[3] M. J. F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," Technical Report TR 291, Cambridge University, 1997.

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker adaptive training," in *ICSLP*, 1996.

[5] J. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Technical Report CMU-CS-94-125, Carnegie Mellon University, March 1994.

[6] B. Kingsbury, G. Saon, L. Mangu, M. Padmanabhan, and R. Sarikaya, "Robust speech recognition in noisy environments: The ibm spine-2 evaluation system," in *ICASSP*, Orlando, Florida, 2002, submitted.

[7] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[8] S. S. Chen and R. A. Gopinath, "Model selection in acoustic modeling," in *Eurospeech*, Budapest, Hungary, Spetember 1999.