

GAUSSIAN MIXTURE MODELING WITH VOLUME PRESERVING NONLINEAR FEATURE SPACE TRANSFORMS

Peder A. Olsen, Scott Axelrod, Karthik Visweswariah and Ramesh A. Gopinath

IBM, T. J. Watson Research Center
134 and Taconic Parkway
Yorktown Heights, NY 10598
{pederao,kv1,axelrod,rameshg}@us.ibm.com

ABSTRACT

This paper introduces a new class of nonlinear feature space transformations in the context of Gaussian Mixture Models. This class of nonlinear transformations is characterized by computationally efficient training algorithms. Experimental results with quadratic feature space transforms are shown to yield modestly improved recognition performance in a speech recognition context. The quadratic feature space transforms are also shown to be beneficial in an adaptation setting.

1. INTRODUCTION

A popular approach to state of the art speech recognition systems uses continuous parameter Hidden Markov Models (HMMs) with the probability density function (pdf) for each state represented by a Gaussian Mixture Model (GMM). Recent investigations has shown that GMMs that model the structure of the quadratic terms of the gaussians more generally than is done, say for diagonal covariance GMMs, can be quite beneficial, [1]. This has driven us to consider higher order nonlinearity. In this paper we suggest incorporating nonlinearity into our models by applying nonlinear feature space transforms. Such a transform is selected by considering likelihood maximization of GMMs in the transformed feature space. In this paper we restrict to the computationally more tractable case when the nonlinear transforms are required to be volume preserving. More specifically, we will require the Jacobian matrix of the transform to be lower triangular as in [2]. Recently, the authors in [3] considered symplectic nonlinear transforms, which are a special case of volume preserving transforms. Actually in [3], the transforms were a special type of symplectic transforms which satisfy the general lower triangular Jacobian matrix condition we impose here. In that paper the nonlinearity was introduced using sigmoid functions, whereas we use quadratic polynomials.

Choosing nonlinear features is problematic particularly

as computational challenges during the training phase can rapidly become insurmountable. Key issues are the computation of the Jacobian matrix, the question of which parameter families to consider and the problem that the number of parameters in general nonlinear feature transforms can become very large. All of these issues are addressed in this paper.

In this paper we will also consider nonlinear feature space adaptation. Feature space adaptation was introduced in a very general way in [4]. Although the formulation there allowed for nonlinear transforms, the only experiments used specialisations of linear transforms. General feature space maximum likelihood linear regression (FM-LLR) transforms was used for adaptation in [5]. We remark that the quadratic model transforms of [6] generalize the linear model transforms of [7] in a way analogous to how the non-linear feature space adaptation transforms here generalize FMLLR transforms. Finally, nonlinear adaptation on a per-dimension basis was considered in [8].

2. MAXIMUM LIKELIHOOD FEATURE SPACE TRANSFORMATIONS

Assume that the input feature vector \mathbf{x} is in \mathbb{R}^d . The goal is to model $\mathbf{y} = \mathbf{f}(\mathbf{x})$ by an HMM, where \mathbf{f} is an *invertible* vector-valued function $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ with $\mathbf{f} = (f_1, \dots, f_d)^T$, and $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$. For data \mathbf{x} at a given HMM state s we wish to model the vector $\mathbf{y} = \mathbf{f}(\mathbf{x})$ by a GMM

$$p(\mathbf{y}|s) = \sum_{g \in s} \pi_g \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \quad (1)$$

The corresponding distribution in the original feature space \mathbf{x} is then of the form

$$\begin{aligned} p(\mathbf{x}|s) &= \frac{\sum_{g \in s} \pi_g \mathcal{N}(\mathbf{f}(\mathbf{x}); \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{|\det(\mathbf{J}(\mathbf{x}))|} \\ &= \sum_{g \in s} \pi_g p(\mathbf{x}|\mathbf{f}, \pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \end{aligned} \quad (2)$$

Here $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of the transform \mathbf{f} ; so the denominator normalizes for volume changes in the \mathbf{x} space.

Using a Viterbi strategy we choose the feature transform \mathbf{f} , the priors π_g , the means $\boldsymbol{\mu}_g$ and the covariances $\boldsymbol{\Sigma}_g$ to maximize the likelihood $\prod_{t=1}^T p(\mathbf{x}_t|s_t)$, where the state sequence (s_1, s_2, \dots, s_T) is the most probable alignment (Viterbi path) of the acoustic training data $\{\mathbf{x}_t\}_{t=1}^T$ to the word transcript. One strategy for maximizing the likelihood is given by the EM algorithm [9]. The EM algorithm introduces an auxiliary function $Q(\Theta, \hat{\Theta})$; where $\Theta, \hat{\Theta}$ denotes model parameters $(\mathbf{f}, \{\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_g)$ and $(\hat{\mathbf{f}}, \{\hat{\pi}_g, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g\}_g)$ respectively. The auxiliary function satisfies $Q(\Theta, \Theta) = 0$ and $L(\Theta) - L(\hat{\Theta}) \geq Q(\Theta, \hat{\Theta})$ where $L(\Theta) = \sum_{t=1}^T \log p(\mathbf{x}_t|s_t)$ is the log likelihood of the training data. The auxiliary function is given by

$$\begin{aligned} Q(\Theta, \hat{\Theta}) &= \sum_{t=1}^T \sum_{g \in s_t} \gamma_{tg} \log \frac{\pi_g p(\mathbf{x}_t | \mathbf{f}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\hat{\pi}_g p(\mathbf{x}_t | \hat{\mathbf{f}}, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)} \\ &= - \sum_g n(g) \ell_g(\Theta), \end{aligned} \quad (3)$$

where γ_{tg} are the occupation counts

$$\gamma_{tg} = \begin{cases} \frac{\hat{\pi}_g p(\mathbf{x}_t | \hat{\mathbf{f}}, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}{\sum_{g^* \in s_t} \hat{\pi}_{g^*} p(\mathbf{x}_t | \hat{\mathbf{f}}, \hat{\boldsymbol{\mu}}_{g^*}, \hat{\boldsymbol{\Sigma}}_{g^*})} & \text{if } g \in s_t \\ 0 & \text{otherwise,} \end{cases}$$

$$n(g) = \sum_t \gamma_{tg} \text{ and}$$

$$\ell_g(\Theta) = - \frac{1}{n(g)} \sum_{t=1}^T \gamma_{tg} \log \frac{\pi_g p(\mathbf{x}_t | \mathbf{f}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\hat{\pi}_g p(\mathbf{x}_t | \hat{\mathbf{f}}, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}. \quad (4)$$

To improve the likelihood $L(\Theta) > L(\hat{\Theta})$ it is sufficient to maximize the auxiliary function $Q(\Theta, \hat{\Theta})$ with respect to Θ . The maximum value with respect to the priors, means and variances of the new model Θ is given by:

$$\begin{aligned} \pi_g &= \frac{n(g)}{\sum_{g^* \in s} n(g^*)}, \\ \boldsymbol{\mu}_{gf} &= \frac{1}{n(g)} \sum_t \gamma_{tg} \mathbf{y}_t \quad \text{and} \\ \boldsymbol{\Sigma}_{gf} &= \frac{1}{n(g)} \sum_t \gamma_{tg} (\mathbf{y}_t - \boldsymbol{\mu}_{gf})(\mathbf{y}_t - \boldsymbol{\mu}_{gf})^T, \end{aligned} \quad (5)$$

where $\mathbf{y}_t = \mathbf{f}(\mathbf{x}_t)$. Using the values above ℓ_g , modulo scaling and constants, becomes

$$\ell_g(\mathbf{f}) = \log |\det \boldsymbol{\Sigma}_{gf}| + \frac{2}{n(g)} \sum_t \gamma_{tg} \log |\det (\mathbf{J}(\mathbf{x}_t))|. \quad (6)$$

The goal is to choose the feature transform \mathbf{f} which maximizes the auxiliary function $Q(\mathbf{f}) = - \sum_g n(g) \ell_g(\mathbf{f})$.

However if there are no constraints on \mathbf{f} beyond invertibility then $\ell_g(\mathbf{f})$ can be made to grow without bound and the optimization problem is ill posed. This problem can be avoided by suitably restricting the feature transforms to a parametric family, say, $\mathbf{f}(\mathbf{x}; \boldsymbol{\phi})$, $\boldsymbol{\phi} \in \mathbb{R}^{n_p}$. The minimum value of $\ell_g(\boldsymbol{\phi}) = \ell_g(\mathbf{f})$ in (6) may then be found using a generic function optimization package. We used the Hilbert Class Library [10] to solve the optimization problems in this paper. The potential problem with this simple and straightforward approach is that the computational cost of evaluating $\ell_g(\boldsymbol{\phi})$ is in general very high. Specifically, ignoring the computation of \mathbf{y}_t , the cost of evaluating $\det(\mathbf{J}(\mathbf{x}_t))$ for $t = 1, \dots, T$ involves $\mathcal{O}(Td^3)$ flops.

2.1. Volume preserving feature space transforms

If we constrain the feature transform family $\mathbf{f}(\mathbf{x}; \boldsymbol{\phi})$ to be volume preserving, $|\det(\mathbf{J}(\mathbf{x}))| = 1$ for all $\mathbf{x} \in \mathbb{R}^d$, then $\ell_g(\boldsymbol{\phi})$ simplifies to

$$\ell_g(\boldsymbol{\phi}) = \log |\det \boldsymbol{\Sigma}_{gf}|, \quad (7)$$

where $\boldsymbol{\Sigma}_f$ is given by (5). In general the computation of $\ell_g(\boldsymbol{\phi})$ requires $\mathcal{O}(d^3 + Td^2)$ flops and T evaluations of \mathbf{f} . One approach to constructing families of volume preserving feature space transforms, used in [2], constrains \mathbf{f} to be such that the matrix $\mathbf{J}(\mathbf{x})$ is lower triangular with ones on the diagonal,

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ f_2(\mathbf{x}) &= x_2 + h_2(x_1) \\ f_3(\mathbf{x}) &= x_3 + h_3(x_1, x_2) \\ &\vdots \\ f_d(\mathbf{x}) &= x_d + h_d(x_1, x_2, \dots, x_{d-1}). \end{aligned} \quad (8)$$

If instead of requiring lower-triangularity of the Jacobian matrix, we merely require that $|\det(\mathbf{J}(\mathbf{x}))|$ is constant with respect to \mathbf{x} we can also consider functions of the form $\mathbf{f}(\mathbf{A}\mathbf{x})$, where $\mathbf{A} \in \mathbb{R}^{d \times d}$. This conveniently makes the ordering of the coordinates irrelevant in (8).

2.2. Affine feature space transform families

A further computational speedup can be achieved by restricting \mathbf{f} to be an affine function of $\boldsymbol{\phi}$

$$\mathbf{f}(\mathbf{x}; \boldsymbol{\phi}) = \mathbf{f}^0(\mathbf{x}) + \sum_{j=1}^{n_p} \phi_j \mathbf{f}^j(\mathbf{x}) = \sum_{j=0}^{n_p} \phi_j \mathbf{f}^j(\mathbf{x}), \quad (9)$$

where $\phi_0 = 1$.

For this family of transforms, sufficient statistics for

computing (7) are:

$$\begin{aligned}\boldsymbol{\mu}^j &= \frac{1}{n(g)} \sum_t \gamma_{tg} f^j(\mathbf{x}_t) \quad \text{and} \\ \boldsymbol{\Sigma}^{ij} &= \frac{1}{n(g)} \sum_t \gamma_{tg} f^i(\mathbf{x}_t) f^j(\mathbf{x}_t)^T - \boldsymbol{\mu}^i (\boldsymbol{\mu}^j)^T.\end{aligned}\quad (10)$$

Computing the statistics (10) costs $\mathcal{O}(Td^2)$ operations and T evaluations of \mathbf{f} , but this computation need only be done once. Consecutive evaluations of $\ell_g(\boldsymbol{\phi})$ can be computed via this statistics using $\boldsymbol{\Sigma}_f = \sum_{i,j=0}^{n_p} \phi_i \phi_j \boldsymbol{\Sigma}^{ij}$. Thus the computation of $\ell_g(\boldsymbol{\phi})$ is reduced to $\mathcal{O}(n_p^2 d^2 + d^3)$ flops. It is important to realize that this cost does not depend on T , the number of training samples.

2.3. Choosing a feature transform

There is still a great amount of freedom in choosing the affine, volume preserving family of transforms. Following the idea of Section 2.2, we will choose the functions h_i in (8) to depend linearly on a parameter $\boldsymbol{\phi}$ as in (9) (and we take $f^0(\mathbf{x}) = \mathbf{x}$). More specifically, we consider the quadratic case where $h_j(x_1, \dots, x_{j-1}) = \sum_{n=1}^{j-1} \sum_{m=1}^n a_{mnj} x_m x_n$, $j = 2, \dots, d$. The number of free parameters is

$$n_p = \binom{d-1}{3} \quad (11)$$

and the corresponding statistics needed to compute (7) is almost the entire set of moment statistics of order ≤ 4 , i.e.

$$\begin{aligned}m_1(a; g) &= \frac{1}{n(g)} \sum_t \gamma_{tg} x_a, \dots \\ m_4(a, b, c, d; g) &= \frac{1}{n(g)} \sum_t \gamma_{tg} x_a x_b x_c x_d.\end{aligned}\quad (12)$$

Taking symmetry into account this statistic consists of $\binom{d}{4} + \binom{d}{3} + \binom{d}{2} + \binom{d}{1}$ unique elements per gaussian. Because of the quartic growth in the size of the statistics we have constrained ourselves to small systems. When estimating the quadratic model we considered only systems with one gaussian per state, although larger systems were occasionally built using the quadratic feature transform from the corresponding acoustic model with one gaussian per state. The feature space dimension was initially set to $d = 20$, and some of the results have been replicated for $d = 39$. It should be noted that an efficient implementation must use all the symmetries and take some care in what order to visit the statistics to avoid excessive cache-misses. There is potentially a speed-up factor of $4! = 24$ for doing this.

2.4. Gradient computation

In order to use the numerical package [10] for the optimization we need to supply the the gradient of the $\ell_g(\boldsymbol{\phi})$ with respect to $\boldsymbol{\phi}$. It can be computed using the chain rule:

$$\frac{\partial \ell_g}{\partial \phi_j} = \text{trace} \left(\boldsymbol{\Sigma}_{gf}^{-1} \frac{\partial \boldsymbol{\Sigma}_{gf}}{\partial \phi_j} \right) = 2 \sum_{i=1}^{n_p} \phi_i \text{trace} \left(\boldsymbol{\Sigma}_{gf}^{-1} \boldsymbol{\Sigma}^{ij} \right).\quad (13)$$

If we add a linear transform then $\ell_g(\boldsymbol{\phi}, \mathbf{A}) = \log |\det \boldsymbol{\Sigma}_{gf}| - 2 \log |\det \mathbf{A}|$ and $\partial \ell_g / \partial A_{ij} = 2(A^{-1})_{ji}$.

3. EXPERIMENTAL RESULTS

Two types of speech recognition experiments were performed with non-linear feature space transforms. In the first set of experiments the transform is used in an acoustic model training setting while in the second set of experiments we consider unsupervised speaker adaptation.

3.1. The test and training databases

The experiments described in this paper was performed on an IBM internal database [11]. Digits are modeled by defining word specific digit phonemes, yielding word models for digits. In total 680 word internal triphones are used to model acoustic context. Two types of acoustic models are considered here. ‘‘Small’’ models have 680 gaussians, one per context dependent state. ‘‘Large’’ models have a total of 10253 gaussians which were distributed across the 680 states using the Bayesian Information Criterion [12]. For initial features 9 consecutive 13 dimensional cepstra vectors were spliced together to yield 117 dimensional vectors. These vectors were subsequently projected into a 20 or 39 dimensional subspace using Linear Discriminant Analysis (LDA) as described in [13]. We constructed full covariance models and Maximum Likelihood Linear Transform (MLLT) models in 20 and 39 dimensions. The covariances in the MLLT case, [14, 15] are constrained to be of the form $\mathbf{B}^{-1} \mathbf{D}_j \mathbf{B}^{-T}$, where $\mathbf{B}, \mathbf{D}_j \in \mathbb{R}^{d \times d}$, \mathbf{D}_j are diagonal matrices and \mathbf{B} are shared over all gaussians. The database used for training consisted of a total of 462388 utterances. The training data was collected in a stationary and moving car at two different speeds – 30 mph and 60 mph. Data was recorded in several different cars with a microphone placed at a few different locations – rear-view mirror, visor and seat-belt. The training data was augmented by synthetically adding noise, collected in a car, to the stationary car data. The test data consists of 22 speakers recorded in a car moving at speeds 0 mph, 30 mph and 60 mph respectively. The total number of words in the test data was 73743. Four tasks were considered: addresses (A), commands (C), digits (D) and radio

control (R). Following are typical utterances from each task:

A: NEW YORK CITY NINETY SIXTH STREET WEST
 C: SET TRACK NUMBER TO SEVEN
 D: NINE THREE TWO THREE THREE ZERO ZERO
 R: TUNE TO F.M. NINETY THREE POINT NINE

3.2. Speech recognition results

The initial feature space in the experiments were either 20 or 39-dimensional. The 20-dimensional feature space was chosen to allow for rapid experimentation. The quadratic feature space transform described in Section 2.3, indicated by $\mathbf{q}(x)$, was used in the experiments. For diagonal covariance GMM’s the objective function $Q(\mathbf{f}) = -\sum_g n(g)\ell_g(\phi)$ must be modified to reflect the diagonal covariance (and the MLLT-transform if present); $\ell_g(\phi) = \log |\det \text{diag} \Sigma_{gf}| - 2 \log |\mathbf{B}|$. Table 1 shows the results for a variety of full covariance and diagonal covariance models, some of which indicate moderate gains in the word error rate (WER). Unfortunately, there was degradation for all but one experiment in the full covariance case.

In the diagonal case the largest gains were seen for the transform $\mathbf{y} = \mathbf{Bq}(\mathbf{x})$ (quadratic transform followed by an MLLT transform) in the 20-dimensional case and for the transform $\mathbf{y} = \mathbf{Bq}(\mathbf{Ax})$ in the 39 dimensional case.

Type	nGauss	Transform	WER	
			$d = 20$	$d = 39$
FCov	680	\mathbf{x}	6.75%	5.13%
		$\mathbf{q}(\mathbf{x})$	6.77%	5.17%
		$\mathbf{q}(\mathbf{Ax})$	6.76%	5.01%
FCov	10K	\mathbf{x}	2.54%	1.71%
		$\mathbf{q}(\mathbf{Ax})$	2.80%	1.73%
Diag	10K	\mathbf{x}	4.14%	3.16%
		$\mathbf{q}(\mathbf{x})$	4.04%	3.05%
		$\mathbf{q}(\mathbf{Ax})$	3.65%	2.76%
MLLT	10K	\mathbf{Bx}	3.78%	2.94%
		$\mathbf{Bq}(\mathbf{x})$	3.56%	2.72%
		$\mathbf{Bq}(\mathbf{Ax})$	3.64%	2.70%

Table 1. Word error rates for full covariance and diagonal covariance models with linear and quadratic feature space transforms.

Further exploring the use of a quadratic feature space transform we considered using a different transform for each HMM state. Table 2 shows the results with 680 and 10K gaussians respectively. In the case of 680 gaussians, where each gaussian has its own quadratic feature transform $\mathbf{q}^j(\mathbf{A}_j\mathbf{x})$, there was a substantial gain over the baseline full covariance model with 680 gaussians. However, the number of parameters needed is quite substantial in this case,

e.g. for $d = 39$ the parameter count is $680n_p \approx 6 \cdot 10^6$ and can be compared to a 10K full covariance system (with $\approx 8 \cdot 10^6$ parameters), whose performance is substantially better. Keeping the state dependent quadratic feature space transforms and training 10K full covariance gaussians we see that the performance is still not very competitive with the 10K full covariance models.

Type	nGauss	Transform(s)	WER	
			$d = 20$	$d = 39$
FCov	680	\mathbf{x}	6.75%	5.13%
		$\mathbf{q}^j(\mathbf{A}_j\mathbf{x})$	4.32%	2.91%
FCov	10K	\mathbf{x}	2.54%	1.71%
		$\mathbf{q}^j(\mathbf{A}_j\mathbf{x})$	2.66%	1.66%

Table 2. Word error rates for full covariance models with state dependent quadratic feature space transforms.

3.3. Adaptation experiments

In this section we report results on several adaptation experiments. In all experiments we performed unsupervised adaptation on 100 test sentences per speaker. The test set consisted of a 147 collections of 100 sentence groups distributed over 22 unique speakers recorded in varying test conditions. In all experiments a first pass decode was done with a baseline model and then adaptation transforms were trained to maximize likelihood under the alignment given by the first pass decode. The results are reported in Table 3. The baseline model, reported on in Table 1, for the first group of experiments uses the 20-dimensional MLLT features \mathbf{Bx} and 10K gaussians. The second group of experiments uses the 20-dimensional nonlinear features $\mathbf{q}(\mathbf{Ax})$ also with 10K gaussians. For each group of experiments we recall the baseline number and report results for adapting with Feature space Maximum Likelihood Linear Regression (FMLLR) [5], i.e. a transform which maps \mathbf{x} to $\mathbf{C}_s\mathbf{x} + \mathbf{b}_s$ for a given speaker s .

Our interest here is in finding what additional gains can be obtained using nonlinear quadratic feature space transforms. There is much less data involved when adapting an acoustic model to a speaker or acoustic environment than when training the acoustic model. Thus the number of parameters that can be adapted should be relatively small. For a speaker specific quadratic feature space transform $\mathbf{q}(\mathbf{x})$, the number of parameters is too large to be supported by the data available for individual speakers in our test set. One way to reduce the number of parameters is to only keep quadratic terms of the form $x_i x_j$ with $i = j$. This makes the number of parameters comparable to the FMLLR case. The speaker specific quadratic transforms we train are of the

form:

$$x_i + b_{si} + \sum_{j=1}^{i-1} W_{ijs} x_j^2, \quad \text{for } i = 1, \dots, d. \quad (14)$$

The third experiment in each of the groups of Table 3 are the results obtained by composing an FMLLR transform with a transform of the form (14).

In both the first group of experiments (with base features \mathbf{Bx}) and in the second group of experiments (with base features $\mathbf{q}(\mathbf{Ax})$), very significant gains were obtained by using the FMLLR transform. Unfortunately, the additional diagonally constrained quadratic transform yields very little additional gain. We do note though that the FMLLR gain was additive with the gain due to the training time transform $\mathbf{q}(\mathbf{Ax})$.

model	Transforms		WER
	FMLLR $\mathbf{u} \rightarrow \mathbf{v}$	nonlinear $\mathbf{v} \rightarrow \mathbf{w}$	
\mathbf{Bx}	\mathbf{u}	\mathbf{v}	3.78%
\mathbf{Bx}	$\mathbf{C}_s \mathbf{u} + \mathbf{b}_s$	\mathbf{v}	2.58%
\mathbf{Bx}	$\mathbf{C}_s \mathbf{u} + \mathbf{b}_s^1$	$b_{is}^2 + \sum_{j=1}^{i-1} W_{ijs}(v_j)^2$	2.53%
$\mathbf{q}(\mathbf{Ax})$	\mathbf{u}	\mathbf{v}	3.65%
$\mathbf{q}(\mathbf{Ax})$	$\mathbf{C}_s \mathbf{u} + \mathbf{b}_s$	\mathbf{v}	2.48%
$\mathbf{q}(\mathbf{Ax})$	$\mathbf{C}_s \mathbf{u} + \mathbf{b}_s^1$	$b_{is}^2 + \sum_{j=1}^{i-1} W_{ijs}(v_j)^2$	2.44%

Table 3. Decoding results for nonlinear feature space adaptation experiments for 10K MLLT GMM acoustic models with $d = 20$.

4. CONCLUSION

We have described a flexible framework in which nonlinear feature transforms can be used in the gaussian mixture modeling framework. For full covariance models we did not see significant gains, and even saw degradations in some cases. For diagonal covariance gaussian mixture model training, as well as for adaptation, we saw modest gains when using nonlinear features. In future research we plan to relax the triangularity constraint on the Jacobian matrix, although this comes at substantial cost during the training phase. Hopefully that will lead to better results.

5. REFERENCES

[1] K. Visweswariah, S. Axelrod, and R. Gopinath, "Acoustic modeling with mixtures of subspace constrained exponential models," in *Proc. Eurospeech*, Geneva, 2003.

[2] J. K. Lin and P. Dayan, "Curved gaussian models with application to modeling of foreign exchange rates," in *Computational Finance - 99*, Y. S. Abu-Mostafa, B. LeBaron, A. W. Lo, and A. S. Weigend, Eds., Cambridge, MA, 1999, MIT Press.

[3] M. K. Omar and M. Hasegawa-Johnson, "Nonlinear maximum likelihood feature transformation for speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, September 2003, vol. 4, pp. 2497–2500.

[4] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE transactions on speech and Audio Processing*, vol. 4, no. 3, pp. 190–202, May 1996.

[5] M. J. F. Gales, "Maximum likelihood linear transformations for HMM based speech recognition," Tech. Rep. TR 291, Cambridge University, 1997.

[6] V. N. Parikh, B. Raj, and R. M. Stern, "Speaker adaptation and environmental compensation for the 1996 broadcast news task," in *Proceedings of the Speech Recognition Workshop*, Chantilly, Virginia, February 1997, DARPA.

[7] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer speech and language*, vol. 9, no. 2, pp. 171–185, 1995.

[8] M. Padmanabhan and S. Dharanipragada, "Maximum likelihood non-linear transformation for acoustic adaptation," *IEEE Transactions on Speech and Audio Processing*, 2003, to appear.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 1977.

[10] M. S. Gockenbach and W. W. Symes, "The Hilbert Class library," <http://www.trip.caam.rice.edu/txt/hcldoc/html/>.

[11] Sabine Deligne, Satya Dharanipragada, Ramesh Gopinath, Benoit Maison, Peder Olsen, and Harry Printz, "A robust high accuracy speech recognition system for mobile applications," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 551–561, November 2002.

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

[13] N. Campbell, "Canonical variate analysis - a general formulation," *Australian Journal of Statistics*, 1984.

- [14] R. A. Gopinath, "Maximum likelihood modeling with gaussian distributions for classification," in *Proceedings of ICASSP*, Seattle, USA, 1998, vol. II, pp. 661–664.
- [15] M. J. F. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions in Speech and Audio Processing*, 1999.