MULTIPLE LINEAR TRANSFORMS

Nagendra Kumar Goel[*]

LSI Logic
C/o Paragea, 207 Perry Parkway
Gaithersburg, MD 20877
ngoel@lsil.com

Ramesh A. Gopinath

IBM T.J. Watson Research Center
Route 134
Yorktown Heights, NY 10598
rameshg@us.ibm.com

ABSTRACT

Recently Heteroscedastic Discriminant Analysis (HDA) has been proposed as a replacement for Linear Discriminant Analysis (LDA) in speech recognition systems that use mixtures of diagonal covariance Gaussians to model the data. Typically HDA and LDA involve a dimension reduction of the feature space. A specific version HDA that involves no dimension reduction, and is popularly known as Maximum Likelihood Linear Transform (MLLT) is often used on the feature space to give significant improvements in performance. MLLT approximately diagonalizes the class covariances, and in effect, tries to approximate the performance of a full-covariance system. However, the performance of a full-covariance system could in some cases be much better than using MLLT-based diagonal covariance system. We propose the method of *Multiple Linear Transforms* (MLT), that bridges this gap in performance, while maintaining the speed efficiency of a diagonal covariance system. This technique improves the performance of a diagonal covariance system, over what could be obtained from HDA or MLLT.

1. INTRODUCTION

We propose a new technique - Multiple Linear Transforms (MLT) - that uses a model-based approach to improve the performance of a diagonal covariance recognition system without incurring the penalty of a full-covariance system. In some cases the performance may even be better than that of the corresponding full-covariance system.

Linear Discriminant Analysis (LDA) [1, 2], and it's generalization Heteroscedastic Discriminant Analysis (HDA) [3, 4], and it's special case, the Maximum Likelihood Linear Transform (MLLT) [6], are now being used in place of augmenting the cepstral features with their first, and second order differences. Gales' global transform for semitied covariance matrices [5] is identical to MLLT but applied in the model space. Demuynck [7] uses a minimum divergence criterion between posterior class distributions in the original and transformed space to estimate an HDA matrix.

Under these approaches, the HDA or LDA techniques are viewed as feature space transformations, that result in a new set of features, which are then used recognition. Saon [8] extensively analyzed various combinations of LDA, HDA, and MLLT, and discovered that the application HDA for feature dimension reduction, followed by MLLT, resulted in the best performance.

As shown by Kumar [4], MLLT can be viewed as a modeling technique that places certain constraints on the gaussian models. However the constraints are too restrictive, since they assume that every gaussian can be diagonalized by using the same linear transformation. Gales' multiple semi-tied covariances relaxes this assumption by assuming that groups of Gaussians have the same diagonalizing transform. The MLT method proposed here relaxes this assumption even further by allowing each gaussian to have its own diagonalizing transform. However, the number of parameters is kept small by allowing tying on a component by component basis.

2. MULTIPLE LINEAR TRANSFORMS

Consider a classification problem where each input feature vector, $x \in \mathcal{R}^n$, has to be classified into one of J classes - with each class modeled by a single full-covariance Gaussian. Under the MLT scheme the input vector x is assigned a class according to the following algorithm. First x is multiplied by a $n \times k$ matrix θ ($k \geq n$), resulting in

$$y = \theta^T x. \tag{1}$$

Each class j has a predefined subset S_j containing exactly n distinct indices in the range $1, \ldots, k$. y_j is defined as the n dimensional vector generated by choosing the subset S_j from y, and $y_{j,l}$ is the l'th component of y_j ($l = 1, \ldots, n$). θ_j is defined as a $n \times n$ submatrix of θ, that is a concatenation of the columns of θ, corresponding to indices in S_j. In addition, $\sigma_{j,l}$ is defined as positive real number denoting the variance of l'th component, of the j'th category.

[*] Author performed work while at IBM T. J. Watson Research Center

Then the combination of θ and S_j defines a unique linear transformation θ_j^T for each class j. Moreover, for a given class, this linear transformation could be potentially different from that for every other class. However, the likelihood of the observation x can still be computed as

$$L_j = 2\log|\theta_j| - \sum_{l=1}^{n} \log \sigma_{j,l} - \sum_{l=1}^{n} \frac{(y_{j,l} - \mu_{j,l})^2}{\sigma_{j,l}}. \quad (2)$$

Then the observation x is assigned to that class, for which, the corresponding value of L_j is maximum. Note that the first two terms in the equation 2 do not depend on x, and therefore, are precomputed, and stored, prior to classification. The parameters of this MLT model are θ, S_j, $\sigma_{j,l}$, and $\mu_{j,l}$. Another view of MLT is that each class is modeled with an *inverse covariance* of the form $\theta_j D_j \theta_j^T$ where columns of θ_j are drawn the columns of θ using S_j and D_j is diagonal[1]. Contrast this with semi-tied covariances where the *covariance* is modeled in the form AD_jA^T (where the A's could be shared across sets of classes and D_j is diagonal).

2.1. Parameter Estimation

We propose estimating the MLT parameters using maximum likelihood method. Given labeled data (x_i, g_i), if θ and S_j are known, then, the remaining parameters are calculated as follows:

$$\mu_j = \frac{\sum_{g_i=j} \theta_j^T x_i}{\sum_{g_i=j} 1}; \quad (3)$$

$$\sigma_j = \frac{\sum_{g_i=j} (\theta_j^T x_i - \mu_j)^2}{\sum_{g_i=j} 1}. \quad (4)$$

Now the likelihood of the data can be written as

$$L(\theta, \{S_j\}) = \sum_{j=1}^{J} N_j * (2\log|\theta_j| - \sum_{l=1}^{n} log(\sigma_{j,l})), \quad (5)$$

where θ_j, and Σ_j implicitly depend on the S_j. S_j is not assumed to be known apriori. Searching for the best S_j (using ML) is infeasible due to the combinatorial complexity. We have investigated two possible heuristics to arrive at θ and corresponding S_j. One is a bottom-up clustering approach, and the other is top-down splitting approach.

2.2. Clustering Approach

For each class j, compute

$$\bar{x}_j = \frac{\sum_{g_i=j} x_j}{\sum_{g_i=j} 1} \quad (6)$$

$$\Sigma_j = \frac{\sum_{g_i=j} (x_j - \bar{x}_j)(x_j - \bar{x}_j)^T}{\sum_{g_i=j} 1} \quad (7)$$

[1]a similar factorization has been reported in [9].

where $\bar{x}_j$ is an $n \times 1$ vector, and Σ_j is a $n \times n$ matrix. let E_j be the matrix containing all the eigenvectors of Σ_j. Let $E_{j,i}$ be the i'th eigenvector of Σ_j. An initial estimate of θ is generated as a $n \times (nJ)$ matrix that is a concatenation of all the eigenvector matrices.

$$\theta = [E_1 \ldots E_J] \quad (8)$$

Also, the initial estimate S_j is given as

$$S_j = \{n(J-1)+1, \ldots nJ\} \quad (9)$$

such that $\theta_j = E_j$. μ_j and σ_j are computed from equations 3 and 4.

We define merging of two directions as the following operations:

1. Two indices o and p are chosen that belong to $\{S_j\}$ and satisfy the constraint that there is no index j for which o, and p both belong to S_j.
2. All the entries in $\{S_j\}$ that equal o, are replaced by the number p.
3. o'th column of θ is removed.
4. 1 is subtracted from all the entries in S_j that are greater than o (to make sure that they still point to the correct column of θ).

Then o is said to be merged to p.

Now new estimate of θ and $\{S_j\}$ are created, by applying the best merge. The best merge is defined as that choice of permissible o and p that results in the minimum reduction in the value of $L(\theta, \{S_j\})$ (see equation 5).

Next, numerical algorithms such as conjugate gradients are used to maximize $L(\theta, \{S_j\})$, with respect to θ. The process of merging and optimization is repeated until, the net decrease in $L(\theta, \{S_j\})$ due to a merge is more than a threshold δ or until θ has the desired size.

2.3. Top down Approach

An alternate scheme would be particularly useful when the pool of directions is small relative to the number of classes. We start with a pool of precisely n directions (recall n is the dimension of the feature space) and estimate the parameters. This is equivalent to estimating the MLLT transform except that there is a parameter S_j, that is initialized to

$$S_j = \{1 \ldots n\} \forall j. \quad (10)$$

We then add another direction to the pool, and randomly assign some S_j entries to this new direction. Then we maximize the likelihood, and reassign the entries in S_j to the direction that results in the maximum gain in likelihood. This procedure is very similar to K means clustering approach, except for the minor difference that instead of the bounded distance function, we are optimizing for the likelihood.

2.4. MLT with Mixture Models and HMMs

Speech Recognition systems, typically employ Hidden Markov Models (HMMs) in which each node, or state, is modeled as a mixture of Gaussians. The well known expectation maximization (EM) algorithm is used for parameter estimation in this case. The techniques described in the previous section easily generalize to this class of models, as follows.

The class index j is assumed to span over all the mixture components of all the states. For example, if there are two states, one with two mixture components, and the other with three, then J is set to five. In any iteration of the EM algorithm, $\gamma_j(t)$ is defined as the probability that the data point at time t belongs to the j'th component. Then equations 3 and 4 are replaced with

$$\mu_j = \frac{\sum_{t=1}^{N} \gamma_j(t)\theta_j^T x_t}{\sum_{t=1}^{N} \gamma_j(t)} \quad (11)$$

$$\sigma_j = \frac{\sum_{t=1}^{N} \gamma_j(t)(\theta_j^T x_t - \mu_j)^2}{\sum_{t=1}^{N} \gamma_j(t)} \quad (12)$$

Similarly, equations 6 and 7 are replaced with

$$\bar{x}_j = \frac{\sum_{t=1}^{N} \gamma_j(t) x_j}{\sum_{t=1}^{N} \gamma_j(t)} \quad (13)$$

$$\Sigma_j = \frac{\sum_{t=1}^{N} \gamma_j(t)(x_j - \bar{x}_j)(x_j - \bar{x}_j)^T}{\sum_{t=1}^{N} \gamma_j(t)} \quad (14)$$

The optimization is then performed as usual, at each step of the EM algorithm.

2.5. Computational Speedup

Under the top down scheme, significant amount of savings in computational cost can be obtained by focusing in one direction at a time. It turns out that given $k-1$ columns of θ the remaining column and the (possibly soft) assignments of training samples to the classes the remaining column of θ can be obtained as the unique solution to a strictly convex optimization problem. This suggests an iterative EM update for estimating θ. The so called Q function in EM for this problem is given by

$$\begin{aligned} Q &= const + \sum_{t,j} \gamma_j(t) log p_j(x_t) \quad (15) \\ &= const - \frac{1}{2}\sum_{t,j} \gamma_j(t) \left\{ -2\log|\theta_j| + \sum_{l=1}^{n} \log|\sigma_{j,l}| \right\} \\ &\quad -\frac{1}{2}\sum_{t,j} \gamma_j(t) \sum_{l=1}^{n} \frac{(y_{j,l}(t) - \mu_{j,l})^2}{\sigma_{j,l}} \end{aligned}$$

Let a be a particular column of θ (i.e. a direction). Let $S(a)$ be the list of states (or classes) that include direction a. Let $|\theta_j^T| = |c_{j,a} a^T|$ where $c_{j,a}$ is the row vector of cofactors associated with complementary (other than a) rows of θ_j^T. Let $\sigma_j(a)$ be the variance of the direction a for state j (i.e., that component of σ_j). Differentiating equation 16 with respect to a (leaving all other parameters fixed) and equating the derivative to zero gives

$$0 = \sum_{j \in S(a),t} \gamma_j(t) \left\{ -2\frac{c_{j,a}}{c_{j,a}a^T} + 2\frac{a}{\sigma_j(a)}(x_t - \bar{x}_j)(x_t - \bar{x}_j)^T \right\} \quad (16)$$

That is,

$$\sum_{j \in S(a),t} \gamma_j(t)\frac{c_{j,a}}{c_{j,a}a^T} = a \sum_{j \in S(a),t} \gamma_j(t)\frac{(x_t - \bar{x}_j)(x_t - \bar{x}_j)^T}{\sigma_j(a)}. \quad (17)$$

Let

$$G = \sum_{j \in S(a),t} \gamma_j(t)\frac{(x_t - \bar{x}_j)(x_t - \bar{x}_j)^T}{\sigma_j(a)}. \quad (18)$$

Then we have the fixed point equation for a

$$a = \sum_{j \in S(a)} \gamma_j \frac{c_{j,a}G^{-1}}{c_{j,a}a^T}, \quad (19)$$

where

$$\gamma_j = \sum_t \gamma_j(t). \quad (20)$$

We suggest a "relaxation scheme" for updating a.

$$a_{new} = \lambda a_{old} + (1-\lambda)\left(\sum_{j \in S(a_{old})} \gamma_j \frac{c_{j,a_{old}}G^{-1}}{c_{j,a_{old}}a_{old}^T} \right), \quad (21)$$

for some $\lambda \in [0, 2]$.

3. EXPERIMENTS AND RESULTS

We conducted experiments on system designed for recognizing command-and-control prompts, in a hands-free car environment. The baseline system uses 39 dimensional features - 13-dim MFCC with first and second order derivatives. The system has 809 context dependent phones. To get a feel for the best possible performance under the MLT framework, we built a context dependent full-covariance system, with 3644 full covariance gaussians. A single MLLT transform was then applied to this system, to obtain a diagonal covariance system. We also applied MLT, with a total of 156 directions ($k = 156$). The results are shown in table 1. As would normally be expected, the best performance is obtained with the full-covariance system. The MLT system performance is better than the MLLT system.

Our next objective was to see if the improvements obtained by MLT hold under various noise conditions. For this

System	Command and Control(CC)	Far Field CC
FullC	4.91	5.15
MLLT	6.29	6.37
MLT(156)	6.06	6.22

Table 1. String Error Rates for the full-covariance, MLLT and MLT systems

System	Near Field Microphone		
	0 mph	30 mph	60 mph
FullC	4.76	4.03	5.70
MLLT	5.68	5.25	7.98
MLT(156)	5.14	4.33	7.07
	Far Field Microphone		
FullC	5.30	4.56	7.75
MLLT	6.29	6.16	12.23
MLT(156)	5.76	5.55	10.41

Table 2. String error rates for FULLC, MLT and MLLT systems, under various microphone and noise conditions

purpose, we trained a new system with 5261 gaussians (and 809 context dependent phones), using multi style training, with the car driving at 30mph, 60mph, and when stationary. The trained system was then tested under various noise conditions, and for both near and far microphone locations, for the similar command and control task. The recognition results are shown in table 2. In this experiment, we find that although a full-covariance system performs the best, MLT improves the performance over MLLT in all the cases.

Figure 1 depicts the increase in likelihood as the number of directions is increased from 39 to 156, for the two systems described above. The exact number of directions that are used in any system, is a design parameter that affects the memory and computational requirements (although marginally). We believe that the performance of a full-covariance system serves as a practical upper bound to the best that can be obtained by using MLT technique, although, in some cases, a better generalization may lead to a performance that is better than the performance of a full-covariance system. It is apparent from the plots that a further increase in likelihood is possible, by increasing the number of directions (k in equation 1). We expect that a few hundred directions is all that would be needed to obtain most of the improvement in performance. These details will be investigated and presented in a future paper.

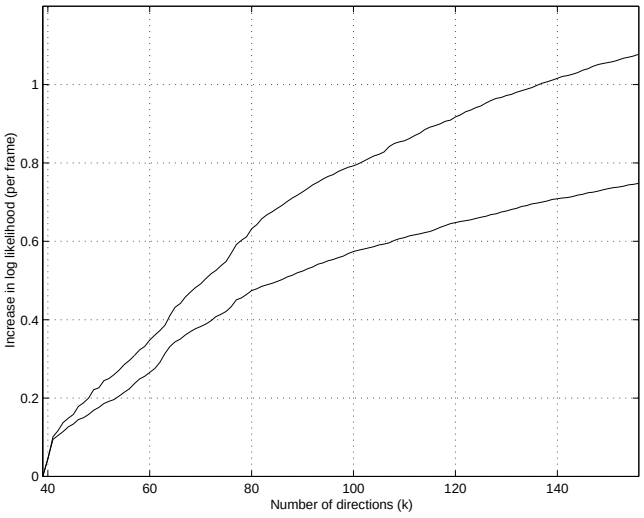

Fig. 1. Log-likelihood increase with # of directions. Top line is for a 5261 gaussian model trained multi-style. Bottom line is for a 3644 gaussian model trained with clean data.

4. REFERENCES

[1] N. A. Campbell. Canonical variate analysis - a general model formulation. *Australian Journal of Statistics*, 26(1):86–96, 1984.

[2] R. O. Duda and P. B. Hart. Pattern classification and scene analysis. *Wiley*, New York, 1973.

[3] Nagendra Kumar. Investigation of Silicon Auditory Models and Generalization of Linear Discriminant analysis for Improved Speech Recognition. *PhD Thesis, Johns Hopkins University*, March 1997.

[4] Nagendra Kumar and A. G. Andreou. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communcation*, 26:283–297, 1998.

[5] M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7:272–281, 1999.

[6] R. A. Gopinath. Maximum likelihood modeling with gaussian distributions for classification. *Proc. ICASSP'98*, Seattle, 1998.

[7] K. Demuynck, J. Duchateau and D. V. Compernolle. Optimal feature sub-space selection based on discriminant analysis. *Proc. EUROSPEECH'99*, Budapest, Hungary, 1999.

[8] G. Saon, M. Padmanabhan, R. Gopinath and S. Chen Maximum Likelihood Discriminant Feature Spaces. *Proc. ICASSP*, pages 1747-1750, 2000.

[9] M. Gales Factored Semi-Tied Covariance Matrices *Proc. NIPS 2000, to appear.*