FLEXIBLE FEATURE SPACES BASED ON GENERALIZED HETEROSCEDASTIC LINEAR DISCRIMINANT ANALYSIS

Alessandro Duminuco, Chaojun Liu, David Kryze, Luca Rigazio

Panasonic Digital Networking Laboratory
5266 Hollyster Avenue, Santa Barbara, CA 93111
duminuco@eurecom.fr {chaojunl,rigazio,kryze}@research.panasonic.com

ABSTRACT

This paper presents a generalized feature projection scheme which allows each feature dimension to be classified in a set of 1 to M classes, where M is the total number of classes. Our method is an extension of the classical full-space null-space approach where each dimension can only be classified in either M classes or 1 class. We believe that this more general formulation allows for a better trade-off of number of parameters versus model complexity, which in turn should provide better classification. We first tested GLDA on TIMIT and obtained an improvement up to 1% in phone classification rate over the best HLDA classifier. Preliminary results on Wall Street Journal 20K also show an improvement over the best HLDA system of about 0.2% absolute.

1. INTRODUCTION

A widely accepted hypothesis in pattern classification is that not all feature dimensions carry enough information to discriminate among all classes. Feature projection addresses this problem by performing a linear transformation of the feature-space and by projecting the feature vectors into a subspace while attempting to preserve the discriminative power. Most projection schemes, including the popular HLDA, estimate the projection transformation by assuming that p dimensions can discriminate among all M classes and the remaining $(n-p)$ have no discriminating power at all. STC ([1]) models the class covariance matrix as a composition of two parts. The first class-dependent and the second shared among a group of classes. SPAM ([2]) models the precision matrix space with a basis superposition, in which the basis matrix ranks can vary freely.

Our method extends these approaches by estimating flexible projection spaces where the discriminating power of each feature dimension is allowed to vary between 1 and M. We derive general formulae for the Maximum-Likelihood estimation of the transformation matrix. Also, we derive two maximization procedures for such a ML solution, one based on gradient-descent and one based on an extension of the HLDA iterative optimization algorithm [3]. We call the resulting Maximum-Likelihood estimation of the flexible projection spaces Generalized Heteroscedastic Linear Discriminant Analysis (GLDA). We believe that GLDA has the potential for improved modeling accuracy because it provides a better trade-off between model parameters and model complexity. We also believe that GLDA provides a strong theoretical framework for the estimation of subspace models [4], used in model compression. In this paper we present the main theoretical results motivating GLDA and experimental results on TIMIT and Wall Street Journal.

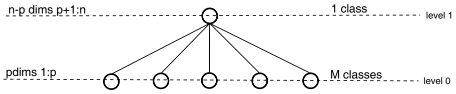

Fig. 1. HLDA feature-space clustering tree.

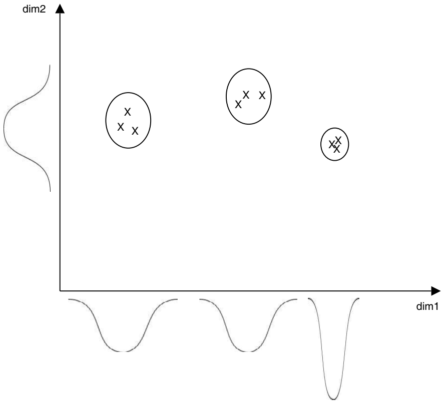

Fig. 2. HLDA clustering plot for $n = 2$, $p = 1$ and $M = 3$.

2. FLEXIBLE FEATURE PROJECTION SPACES

HLDA is the application of the EM algorithm ([5], [6], [7]) to the problem of feature projection. Given the observation vector $x \in \Re^n$ belonging to M classes, HLDA assumes that the first p dimensions (the full-space) can discriminate among the M classes and models them with M gaussian distributions, while the last $(n-p)$ dimensions (the null-space) are assumed to have no discriminating power and are modeled by a one shared distribution [8]. In essence HLDA performs a feature-space tieing over the last $(n-p)$ dimensions across all classes. We can graphically represented this with the feature-space clustering tree of figure 1 and, as an illustrative example, with the graph of figure 2. HLDA estimates the transformation matrix that maximizes the likelihood given such a feature-space tieing structure.

The basic idea of GLDA is to allow for the construction of a more general feature-space clustering tree which has more than two levels and with a variable number of classes at the different levels of the tree [9]. GLDA allows the discriminating power of each dimension to vary between 1 and M. Figure 3 shows an example of

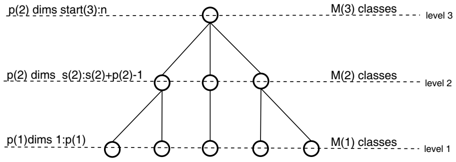

Fig. 3. GLDA feature-space clustering tree.

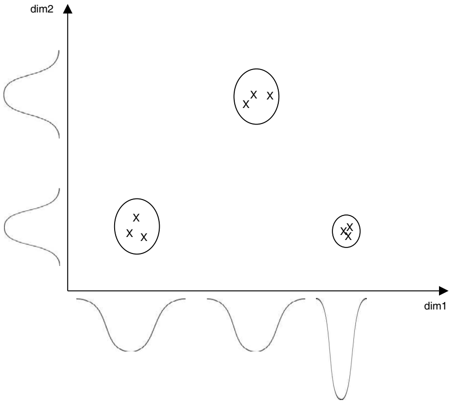

Fig. 4. GLDA clustering plot for $n = 2$, $p(1) = 1$, $p(2) = 1$, $M(1) = 3$ and $M(2) = 2$.

such a generalized feature-space clustering tree and figure 4 shows an example of data distribution which could profit from that.

3. GENERALIZED FEATURE-SPACE CLUSTERING

Notice that the structure of the feature-space clustering tree is reflected in the parameters structure. In the case of HLDA, for a class m the mean vector $\hat{\mu}_m$ and covariance matrix $\hat{\Sigma}_m$ in the transformed space are partitioned into two blocks: the first block of p dimensions depends on the class, and has parameters $\hat{\mu}_{[p],m}$ and $\hat{\Sigma}_{[p],m}$, while the second block of $(n-p)$ dimensions does not depend on the class and is shared among all classes with parameters $\hat{\mu}_{[n-p]}$ and $\hat{\Sigma}_{[n-p]}$:

$$\hat{\mu}_m = \begin{bmatrix} \hat{\mu}_{[p],m} \\ \hat{\mu}_{[n-p]} \end{bmatrix} \quad \hat{\Sigma}_m = \begin{bmatrix} \hat{\Sigma}_{[p],m} & 0 \\ 0 & \hat{\Sigma}_{[n-p]} \end{bmatrix}$$

The ML estimate of the transformation matrix A^* is obtained by maximizing the Q-function, which expresses the likelihood in the transformed space:

$$A^* = \arg\max_A Q(\hat{\Sigma}_m)$$

where the Q has the following expression:

$$Q = \log|A| - \frac{1}{2}\sum_m \frac{\gamma(m)}{\gamma} \log|\hat{\Sigma}_m|$$

where $\gamma(m)$ is the posterior for class m and γ is the total count. By exploiting the block-diagonal structure of the covariance matrix, the determinant of the HLDA covariance can be written as:

$$|\hat{\Sigma}_m| = |\hat{\Sigma}_{[p],m}||\hat{\Sigma}_{[n-p]}|$$

By using the linear projection rule and by extracting the rows of the transformation matrix corresponding to each block, the previous expression becomes:

$$|\hat{\Sigma}_m| = |A_{1:p}\Sigma_m A'_{1:p}||A_{p+1:n}\Sigma A'_{p+1:n}|$$

where Σ_m and Σ are the covariance matrices referring to the original non-transformed space. Moreover Σ is the global covariance matrix computed over all the classes. Notice that this factorization corresponds to having one term of the Q-function for each node of the feature-space clustering tree, where the rows of the transformation matrix are indexed by the level of the tree and the class of covariance matrix is indexed by the leaves of the sub-tree.

In the case of GLDA this structure is generalized to a covariance matrix of L blocks and the associated mean vectors:

$$\hat{\mu}_m = \begin{bmatrix} \hat{\mu}_{[p(1)],m_1} \\ \hat{\mu}_{[p(2)],m_2} \\ \hat{\mu}_{[p(3)],m_3} \end{bmatrix}$$

$$\hat{\Sigma}_m = \begin{bmatrix} \hat{\Sigma}_{[p(1)],m_1} & 0 & 0 \\ 0 & \hat{\Sigma}_{[p(2)],m_2} & 0 \\ 0 & 0 & \hat{\Sigma}_{[p(3)],m_3} \end{bmatrix}$$

We can then compute the determinant of the GLDA covariance matrix in the transformed space as the product of its block determinants:

$$|\hat{\Sigma}_m| = \prod_{l=1}^{L} |\hat{\Sigma}_{[p(l)],m_l}|$$

where L is the total number of levels and $p(l)$ is the number of dimensions associated with level l. Also notice that each covariance block is computed selectively using rows of A. By applying the same rules on the generalized feature-space clustering tree, the GLDA Q-function can be written as:

$$Q = \log|A| - \tfrac{1}{2}\textstyle\sum_{l=1}^{L}\sum_{m_l=1}^{M_l}\{$$
$$\tfrac{\gamma(m_l)}{\gamma} \log|A_{[s(l):s(l)+p(l)]-1}\Sigma_{\{m:m\epsilon m_l\}}A'_{[s(l):s(l)+p(l)-1]}|\}$$

where $M(l)$ is the number of classes associated with level l and $s(l)$ is the initial dimension for level l, defined by the recursion $p(l)$: $s(l) = s(l-1) + p(l-1)$ with $s(1) = 1$.

4. MAXIMUM-LIKELIHOOD OPTIMIZATION

Two key issues that need to be addressed to estimate the flexible feature spaces: the maximization of the Q-function and the construction of the feature-space clustering tree. As for most other feature transformation methods (with the noticeable exception of MLLU [10]) the maximization of the Q-function does not have any close-form solution and requires numerical optimization. The first method explored is based on gradient descent, while the second and most effective solution is based on a generalized version of the iterative maximization originally derived for HLDA. Gradient methods, such as steepest descend or conjugate gradient, require first derivative of the objective function. For GLDA the first derivative of the Q-function is:

$$\frac{\partial Q}{\partial A} = A'^{-1} - \begin{bmatrix} D_1 \\ \vdots \\ D_l \\ \vdots \\ D_L \end{bmatrix}$$

where the D_l is a $p(l) \times n$ matrix relative to the level l in the feature-space clustering tree and its expression is:

$$D_l = \sum_{m_l=1}^{M_l} \frac{\gamma(m_l)}{\gamma} \{ \\ (A_{[s(l):s(l)+p(l)-1]} \Sigma_{\{m:m \epsilon m_l\}} A'_{[s(l):s(l)+p(l)-1]})^{-1} \\ A_{[s(l):s(l)+p(l)-1]} \Sigma_{\{m:m \epsilon m_l\}} \}$$

For our experiments we used a standard conjugate gradient optimization routine provided by the GSL package.

Regarding the iterative optimization, it was originally proposed for STC [1] and then adapted for HLDA in [3] and relies on the assumption of diagonal covariance matrices in the projected space. The extension of this method for GLDA follows the same iterative scheme:

1. Initialization: The matrix A is initialized by LDA.
2. Parameter Projection: Using the current estimate of A, the means $\hat{\mu}_m$ and the diagonal covariance matrices $\hat{\Sigma}_m^{Diag}$ are computed.
3. Projection Re-estimation: Using the current parameters in the projected space, a new transformation A is estimated in a way guaranteed to improve the likelihood.
4. Iteration: Steps 2 and 3 are repeated until convergence is reached (usually requires less than 10 iterations).

Step 2 performs a projection in which the off-diagonal elements of the covariance matrices are nulled out. Step 3 is in itself an iterative process, which estimates the new transformation matrix row by row. The i-th row of A is estimated with the following expression:

$$a_i = c_i G_i^{-1} \sqrt{\frac{\gamma}{c_i G_i^{-1} c_i'}}$$

where c_i is the i-th row of the cofactors of the matrix A and G_i is the auxiliary function defined as:

$$G_i = \begin{cases} \sum_{m=1}^{M} \frac{\gamma(m)}{\hat{\sigma}_{m,i}^{Diag}} \Sigma_m & i \leq p \\ \\ \frac{\gamma}{\hat{\sigma}_i^{Diag}} \Sigma & i > p \end{cases}$$

It can be demonstrated that generalizing the method to GLDA only requires a new auxiliary function, which can be written as:

$$G_i = \sum_{m_l=1}^{M_l} \frac{\gamma(m_l)}{\hat{\sigma}_{m_l,i}^{Diag}} \Sigma_{\{m:m \epsilon m_l\}}$$

4.1. Feature-Space Clustering Tree Estimation

Our derivation assumes that the structure of the feature-space clustering tree is available before the optimization procedure is started. However deriving a good structure for the feature clustering tree is non trivial. We need to address two different issues: the definition of a good tree structure, i.e. the number of classes and the number of dimensions for each level, and the definition the super-classes corresponding to the internal nodes.

For the first issue we used a technique proposed in [8], which is essentially based on cross-validation: the structure defined by {L,M(l),p(l)} is explored using a first set of the training data and is validated with an independent set. The structure which provides the best results is used for the ML optimization.

Regarding the definition of the super-classes we explored two different approaches: The first is based on HMM topological information, while the second is based on unsupervised clustering.

1. The topological clustering associates each class to a single mixture component in a state and construct the super-classes by mapping components to states, and states to phonetic units. This does not require any optimization as the topology of the HMM is already available, is sub-optimal.
2. The second method is based on minimum likelihood loss bottom-up clustering. This is a greedy merging of two classes based on the likelihood loss computed with full class covariance matrices:

   $$\gamma_1 \log \frac{|\Sigma_{1,2}|}{|\Sigma_1|} + \gamma_2 \log \frac{|\Sigma_{1,2}|}{|\Sigma_2|}$$

   where $\Sigma_1, \Sigma_2, \gamma_1, \gamma_2$ are the covariance matrix and the posterior of two classes to be merged, and $\Sigma_{1,2}$ is the resulting merged covariance matrix, all covariances considered in the original non projected space. This method provides better performance but is quadratic in the number of classes.

Other unsupervised clustering techniques should be explored, especially the top-down clustering, which should address the complexity issue of the bottom-up clustering.

5. EXPERIMENTS

Initial validation experiments were conducted on TIMIT. We selected phonemes out of the TIMIT database and used the hand labels to extract the frames associated to each phone. Feature vectors were computed using MFCC with first and second derivatives, yielding a total of 39 dimensions. We then performed phone classification experiments using one gaussian per phone-class and we compared the performance of the HLDA and the GLDA transformation matrices.

The results provided by HLDA are reported in figure 5. The best result is obtained with the iterative optimization and $p = 11$. We then ran GLDA with a three level tree, in which the number of dimensions associated with the first level is near the optimal value of p obtained with HLDA. We then added two more levels to the feature-space clustering tree and optimized the feature-space with GLDA. The results are reported in table 1 and show that GLDA improves the best HLDA score of up to 1%.

$p(1)$	$p(2)$	$M(2)$	HLDA Baseline	GLDA Rate
11	15	3	79.38%	80.29%
13	5	9	78.36%	80.18%
15	5	3	78.93%	79.84%
18	5	3	78.70%	79.04%

Table 1. Phone classification results on TIMIT.

We then tested GLDA on the Wall Street Journal 20K words dictation task. The baseline system is based on a MFCC front-end

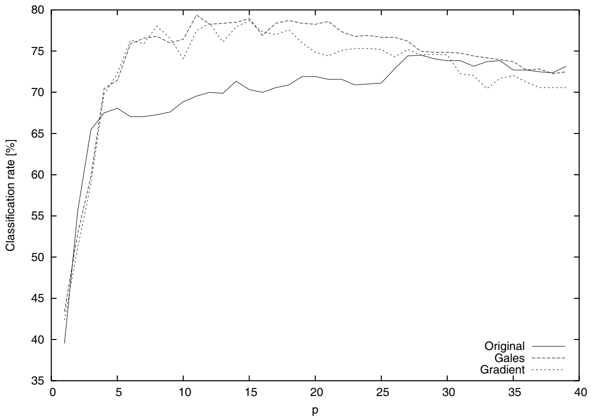

Fig. 5. HLDA phone classification results on TIMIT data. The three curves represent respectively the classification rate in the original space, with HLDA transformation computed with gradient and with the iterative methods. All curves are of the dimension of the projected space.

estimated on a window size of 20 milliseconds and with a frame shift of 10 milliseconds. First and second derivatives are then computed providing a total feature vector size of 39, and cepstral mean normalization is applied at the sentence level. The acoustic models are gender-independent word-internal tree-clustered triphones models with 2245 states and a total of 64000 gaussians. The acoustic models are trained on the WSJ0 and the WSJ1 training sets which provide a total of about 60 hours of speech. The recognition system is based on a one-pass trigram decoder [11] and gives a baseline word recognition accuracy of 88.73%. When applying MLLU feature transformation we obtain a word accuracy of 88.98%. HLDA is tested with two different values of p, achieving a score of 89.51% for $p = 33$ and 89.74% for $p = 36$. For GLDA we use topological clustering, with each gaussian representing a different class for the tree leaves and each state representing a super-class in the second level of the feature-space clustering tree. As for the TIMIT experiment, GLDA is run with $p(1) = p$ and for some different values for $p(2)$. Table 2 report the results obtained with HLDA and GLDA:

$p(1)$	$p(2)$	$M(2)$	HLDA baseline	GLDA rate
33	3	2245	89.51%	89.60%
	5			89.70%
36	2		89.74%	89.76%

Table 2. Recognition results (word accuracy) on the Wall Street Journal 20K words dictation task.

The results on Wall Street Journal show that GLDA provides an improvement for a large vocabulary task over the best HLDA baseline, even if it is somewhat limited. However notice that 64000 classes are probably not enough to generate a over-training for HLDA and GLDA is likely to have an advantage for larger acoustic models or multiple projection schemes.

6. CONCLUSIONS

We proposed a method to estimate flexible feature projection spaces which extends the current formalism of full-space null-space used in many feature projection methods. The flexibility of our method provide increased robustness to over-training by allowing a more flexible trade-off between model parameters and model resolution. Experiments show that our method can always improve the best HLDA performance, provides better classification rate (on TIMIT, improvement of up to 1%) and improves word accuracy (on Wall Street Journal 20K, improvement of 0.2%). While showing a good potential, our investigation is still preliminary, as issues such as the choice of the feature-space clustering tree and the complexity of the estimation of flexible feature transformation have only been partially addressed.

7. REFERENCES

[1] M.J.F. Gales, "Semi-tied Covariance Matrices for Hidden Markov Models," Tech. Rep., Cambridge University, 1998.

[2] R. Gophinat S. Axelrod and P. Olsen, "Modeling with a subspace constraint on inverse covariances matrices," in *Proc. of ICSLP*, 2002.

[3] M.J.F. Gales, "Maximum Likelihood Multiple Projection Schemes for Hidden Markov Models," Tech. Rep., Cambridge University, 1998.

[4] B. Mak and E. Bocchieri, "Training of subspace distribution clustering hidden markov model," 1998.

[5] 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 B*, pp. 1–38, 1977.

[6] M.J.F. Gales, "Maximum Likelihood Linear Transformation for HMM-Based Speech Recognition," Tech. Rep., Cambridge University, 1998.

[7] R. Gophinat, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. of ICASSP*, 1998.

[8] N. Kumar, *Investigation in Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, Ph.D. thesis, John Hopkins University, New York, 1997.

[9] A. Duminuco, "Generalized LDA. An improved projection scheme for speech recognition," M.S. thesis, Institut Eurecom, Sophia Antipolis, France, September 2005.

[10] Patrick Nguyen, Luca Rigazio, Christian Wellekens, and Jean-Claude Junqua, "Lu factorization for feature transformation," in *Proc. of ICSLP*, 2002.

[11] P. Nguyen, L. Rigazio, and J.-C. Junqua, "EWAVES: an efficient decoding algorithm for lexical tree based speech recognition," in *Proc. of ICSLP*, Beijing, China, Oct. 2000, vol. 4, pp. 286–289.