MINIMUM PHONEME ERROR BASED HETEROSCEDASTIC LINEAR DISCRIMINANT ANALYSIS FOR SPEECH RECOGNITION

Bing Zhang † *and Spyros Matsoukas*

BBN Technologies, 50 Moulton St. Cambridge, MA 02138
{bzhang, smatsouk}@bbn.com

ABSTRACT

In this paper we introduce a discriminative feature analysis method that seeks to minimize phoneme errors in lattice-based training frameworks. This technique, referred to as Minimum Phoneme Error Heteroscedastic Linear Discriminant Analysis (MPE-HLDA), is shown to be more robust than traditional LDA methods in high dimensional spaces, and easy to incorporate with existing training procedures, such as HLDA-SAT and discriminative training of Hidden Markov Models (HMMs). Results on conversational telephone speech and broadcast news corpora also show that the recognition accuracy is improved using features selected by MPE-HLDA.

1. INTRODUCTION

In speech recognition systems, feature analysis is usually employed for better classification accuracy and complexity control. In recent years, extensions to the classical Linear Discriminant Analysis (LDA) have been widely adopted. Among them, Heteroscedastic Discriminant Analysis (HDA) [1] seeks to remove the equal class variance constraint assumed by LDA. In addition, the authors of [1] applied a Maximum Likelihood Linear Transformation (MLLT) on top of LDA and HDA and improved accuracy. Maximum likelihood based Heteroscedastic Linear Discriminant Analysis (HLDA) [2, 3], generalizes LDA in another way by putting the optimization of feature projections inside an ML parametric estimation framework, taking the HMM structure (e.g. diagonal covariance Gaussian mixture state distributions) into consideration.

Despite the differences between the above techniques, they have some common limitations. First, none of them assumes any prior knowledge of confusable hypotheses, so their choices are determined to be suboptimal for recognition. Second, their objective functions do not directly relate to the word error rate (WER), which is often the performance measure of speech recognition systems. As a result, it is often unknown whether selected features will do well in testing by just looking at the values of objective functions. For example, we found that HLDA could select totally non-discriminant features while improving its objective function by mapping all training samples to a single point in space along some dimensions.[1]

Recent research showed that using discriminative criteria like Maximum Mutual Information (MMI) [4] and Minimum Phoneme Error (MPE) [5] in optimizing Gaussian parameters improved recognition accuracy significantly. Inspired by this work, we developed a feature analysis approach based on the MPE criterion, MPE-HLDA, for better recognition accuracy. In addition, since this criterion is closely related to WER, MPE-HLDA tends to be more robust than other projection methods, which makes it potentially better suited for a wider variety of features.

Under the MPE criterion, we could perform EM updates of feature projections and Gaussian parameters jointly [6], however, this would tie projections closely with discriminatively placed Gaussians. Instead, we would like the optimization to concentrate on finding better features by running the optimization of feature projections as a standalone process. By doing this we can have more flexibility in using the resulting projections. For instance, we could perform regular ML training with them and still preserve the gains from MPE-HLDA. We have also been able to obtain improvements from MPE-HLDA on top of HLDA-SAT training [7], as we will see in a later section of this paper.

More details of MPE-HLDA will be covered in sections 2 and 3. In section 4, the system configuration of our experiments on conversational telephone speech (CTS) and broadcast news (BN) corpora is described. The results are then compared and analyzed in section 5. Finally the paper ends with conclusions and suggestions for future research.

2. MPE OBJECTIVE FUNCTION AND DERIVATIVE

If we define $A_{p\times n}$ as a global feature projection matrix that linearly maps n-dimensional original features to p-dimensional ones, where $p < n$, then Gaussian parameters and features in the p-dimensional space can be written as

$$\hat{\mu}_m = A\,\mu_m \quad (1)$$

$$\hat{C}_m = diag(A\Sigma_m A^T) \quad (2)$$

$$\hat{o}_t = A\,o_t \quad (3)$$

where μ_m and Σ_m are the mean and full covariance of Gaussian m in the original feature space. We will refer to $\lambda = \langle \hat{C}_m, \hat{\mu}_m \rangle$, the model in reduced feature space, as the MPE-HLDA model.

MPE-HLDA aims at minimizing expected number of phoneme errors introduced by the MPE-HLDA model in a given hypothesis lattice, or equivalently maximizing the function

$$F_{MPE}(\hat{O}, \lambda) = \sum_{r}^{R} \sum_{w_r} P_\lambda(w_r \mid \hat{O}_r)\, \varepsilon(w_r) \quad (4)$$

where R is the total number of training utterances, $\hat{O}_r$ is the sequence of p-dimensional observation vectors in utterance r, and

† Bing Zhang is a Ph.D. student at the College of Computer and Information Science, Northeastern University, Boston, MA 02115

[1] This can easily occur when the original feature space contains features that are linearly dependent in some dimensions.

$\varepsilon(w_r)$ is the "raw accuracy" score of word hypothesis w_r as defined in [5]. $P_\lambda(w_r \mid \hat{O}_r)$ is the posterior probability of hypothesis w_r in the lattice, computed as follows

$$P_\lambda(w_r \mid \hat{O}_r) = \frac{P_\lambda(\hat{O}_r \mid w_r)^k P(w_r)}{\sum_{w'_r} P_\lambda(\hat{O}_r \mid w'_r)^k P(w'_r)} \tag{5}$$

where $P(w_r)$ is the language model probability of hypothesis w_r, and k is an exponent applied to the acoustic scores in order to reduce their dynamic range, thereby avoiding the concentration of all posterior mass in the top-1 hypothesis of the lattice.

Note that $P_\lambda(\hat{O}_r \mid w_r)$ is the regular HMM observation probability of the projected features $\hat{O}_r$ given the lattice hypothesis w_r, computed based on the MPE-HLDA model λ.

It can be shown that the derivative of (4) with respect to A is

$$\frac{\partial F_{MPE}(\hat{O}, \lambda)}{\partial A} = k \sum_r^R \sum_{q_r} \mathcal{D}(q_r, r) \frac{\partial \log P_\lambda(\hat{O}_{q_r} \mid q_r, \lambda)}{\partial A} \tag{6}$$

where

$$\mathcal{D}(q_r, r) = P_\lambda(q_r \mid \hat{O}_{q_r}, \lambda) \left[\varepsilon'(q_r) - \alpha'(r)\right] \tag{7}$$

$\alpha'(r)$ is the MPE score of utterance r and $\varepsilon'(q_r)$ is the average accuracy over all hypotheses that contain arc q_r. It has been shown in [5] that both $\varepsilon'(q_r)$ and $\alpha'(r)$ can be computed efficiently via forward-backward passes over error-marked lattices.

Within arc q_r, we also have

$$\frac{\partial \log P_\lambda(\hat{O}_{q_r} \mid q_r, \lambda)}{\partial A} = \sum_{t=S_{q_r}}^{E_{q_r}} \sum_m \gamma_{q_r}^m(t) \frac{\partial \log P_\lambda(\hat{o}_t \mid m, \lambda)}{\partial A} \tag{8}$$

where S_{q_r} and E_{q_r} are the begin and end time of arc q_r, respectively, and $\gamma_{q_r}^m(t)$ denotes the posterior probability of Gaussian m in arc q_r at time t.

Finally, the derivative of log Gaussian likelihood with respect to A in (8) is

$$\frac{\partial \log P_\lambda(\hat{o}_t \mid m, \lambda)}{\partial A} = \hat{C}_m^{-1} \left(\hat{C}_m^{-1} P_t^m - I_p\right) A \Sigma_m - \hat{C}_m^{-1} R_t^m \tag{9}$$

where

$$P_t^m = diag\left[(\hat{o}_t - \hat{\mu}_m)(\hat{o}_t - \hat{\mu}_m)^T\right] \tag{10}$$

$$R_t^m = (\hat{o}_t - \hat{\mu}_m)(o_t - \mu_m)^T \tag{11}$$

and I_p denotes the $p \times p$ identity matrix. Therefore, Eq. (6) can be rewritten as

$$\frac{\partial F_{MPE}(\hat{O}, \lambda)}{\partial A} = k \sum_m \hat{C}_m^{-1} \left(\hat{C}_m^{-1} \mathcal{G}_m - \zeta_m I_p\right) A \Sigma_m - k\mathcal{J} \tag{12}$$

where

$$\zeta_m = \sum_r \sum_{q_r} \mathcal{D}(q_r, r) \sum_{t=S_{q_r}}^{E_{q_r}} \gamma_{q_r}^m(t) \tag{13}$$

$$\mathcal{G}_m = \sum_r \sum_{q_r} \mathcal{D}(q_r, r) \sum_{t=S_{q_r}}^{E_{q_r}} \gamma_{q_r}^m(t) P_t^m \tag{14}$$

$$\mathcal{J} = \sum_m \hat{C}_m^{-1} \sum_r \sum_{q_r} \mathcal{D}(q_r, r) \sum_{t=S_{q_r}}^{E_{q_r}} \gamma_{q_r}^m(t) R_t^m \tag{15}$$

3. MPE-HLDA IMPLEMENTATION

Calculation of the original space statistics, μ_m and Σ_m, is done by running a forward-backward pass over the training data using a model trained with ML in the reduced feature space provided by the initial estimate of A. The gaussian posterior probabilities computed in the backward pass are used to accumulate sufficient statistics for μ_m and Σ_m in the original feature space. This technique is usually referred to as "single pass retraining". To limit the amount of memory required during accumulation, the entire process is divided into a number of smaller jobs, each accumulating statistics for a subset of the Gaussians.

In theory, the derivative of the MPE-HLDA objective function can be computed based on Eq. (12), via a single forward-backward pass over the training lattices. In practice, however, it is not possible to fit all the full covariance matrices Σ_m in memory, since most of the state of the art HMMs use a large number of Gaussians. One could try to access the original space statistics from the secondary storage during the forward-backward training, but this would severely reduce the efficiency of the process.

Instead, we compute the derivative in two steps. First, we run a forward-backward pass over the training lattices to accumulate ζ_m, $\mathcal{G}_m$ and $\mathcal{J}$, and then have another step that uses these statistics together with the full covariance matrices Σ_m to synthesize the derivative.

Based on the consideration of flexibility discussed in section 1, we used gradient descent in updating the projection matrix, instead of doing EM updates of all model parameters. Though commonly thought to be inefficient, in practice we found that the gradient descent optimization usually converges in less than 20 iterations.

Once we have optimized the MPE-HLDA model, we can use it for a second retraining pass, to update μ_m and Σ_m. This step is especially useful if the initial projection and ML model are far from the optimal configuration. The result is an iterative optimization procedure, as follows:

1. Initialize feature projection matrix $\mathcal{A}^{(0)}$ by LDA or HLDA, and MPE-HLDA model $\hat{\mathcal{M}}^{(0)}$ by Gaussian splitting.
2. Set $i \leftarrow 1$.
3. Compute covariance statistics $\Sigma_m^{(i)}$ in the original feature space:
   (a) Do maximum likelihood update of MPE-HLDA model $\hat{\mathcal{M}}^{(i-1)}$ in the feature space defined by $\mathcal{A}^{(i-1)}$.
   (b) Do single pass retraining using $\hat{\mathcal{M}}^{(i-1)}$ to generate $\mu_m^{(i)}$ and $\Sigma_m^{(i)}$ in the original feature space.
4. Optimize the feature projection matrix:
   (a) Set $j \leftarrow 0$, $A_j^{(i)} \leftarrow \mathcal{A}^{(i-1)}$.
   (b) Project $\mu_m^{(i)}$, $\Sigma_m^{(i)}$ using $A_j^{(i)}$ to get model $\hat{\mathcal{M}}_j^{(i)}$ in reduced subspace.
   (c) Run forward-backward pass on lattices using $\hat{\mathcal{M}}_j^{(i)}$ to compute ζ_m, $\mathcal{G}_m$ and $\mathcal{J}$.
   (d) Use $\Sigma_m^{(i)}$, $A_j^{(i)}$ and statistics from 4c to compute the MPE objective function and its derivative.
   (e) Update $A_j^{(i)}$ to $A_{j+1}^{(i)}$ using gradient descent.
   (f) Set $j \leftarrow j + 1$, goto 4b unless convergence or maximum number of iterations is reached.

5. Optionally, set $\mathcal{A}^{(i)} \leftarrow A_{j-1}^{(i)}$, $\hat{\mathcal{M}}^{(i)} \leftarrow \hat{\mathcal{M}}_{j-1}^{(i)}$, $i \leftarrow i+1$ and go to 3 to repeat.

It should be noted that pre-computing and storing μ_m and Σ_m in the above procedure could be very expensive both in terms of computation and storage when the dimensionality of the features in the original space is very high. Fortunately, there is a way to rearrange the terms in Eq. (12) such that they do not rely on sufficient statistics of the original space, leading to a "memory efficient" implementation of MPE-HLDA. This implementation, however, requires additional forward-backward passes over the training data in each iteration of MPE-HLDA in order to accumulate sufficient statistics of the reduced space for the derivative calculation.

4. EXPERIMENTAL SETUP

We ran experiments on both Conversational Telephone Speech (CTS) and Broadcast News (BN) corpora, as part of the DARPA EARS research project [8]. For CTS, we have approximately 2300 hours of training data, of which 800 hours were used for training the initial ML model $\hat{\mathcal{M}}^{(0)}$ and the remaining were treated as held-out training data for lattice generation and discriminative training. For MPE-HLDA, only 370 hours of the held-out data were used, in order to reduce training time. Similarly for BN, we used 600 hours of data from Hub4 and TDT corpora for training the initial model $\hat{\mathcal{M}}^{(0)}$ and 330 hours of held-out data for MPE-HLDA estimation.

Experiments were performed both with and without HLDA-SAT. With HLDA-SAT, a CMLLR-based adaptation was performed first on 15-dimensional (14 cepstral coefficients and normalized energy) Perceptual Linear Predictive (PLP) coefficients, resulting in adapted cepstra with smaller within-class variances, which served as input to MPE-HLDA. Besides the cepstral difference, HLDA-SAT is totally transparent to MPE-HLDA.

Our first experiments used cepstra and their first, second and third derivatives as input feature vectors for HLDA. Recent research, however, suggested that there is more useful information in longer contexts (concatenated frames), especially for the CTS corpus. To reduce the computational cost, we adopted a "two-level projection" approach. Given dimensionality of concatenated features ℓ and target dimensionality p, we first estimate a projection matrix $L_{n\times\ell}$ on the first level and keep it fixed throughout the MPE-HLDA optimization. On the second level, we use MPE-HLDA to update the projection matrix $A_{p\times n}$ using the procedure described in sections 2 and 3. In usage, $A \cdot L$ forms the global feature projection. ℓ is determined by the number of frames to concatenate. p and n are chosen based on experience, typically $p = 60$ and $n = 130$ or 190 in our experiments.

For acoustic modeling, State Cluster Tied Mixtures (SCTM) were used. The SCTM model uses two levels of state tying; at the first level, states are tied based on a decision tree to share Gaussian parameters, while at the second level states share mixture weights for a particular Gaussian cluster (codebook). Each MPE-HLDA model contained only 12 components for each codebook cluster, much smaller than in regular systems, and it was used only for estimating the feature projection. The initial model was trained based on the ML criterion with labeled data and an initial feature projection, which was formed by a first-level LDA projection and a second-level HLDA projection.

Lattices were generated on held-out training data, and marked with trigram language model scores and arc accuracy scores [5].

Iteration	MPE-HLDA model		Final ML model
	tr. WER	Eval03	Eval03
0 (HLDA)	30.5	30.6	25.9
1 (MPE-HLDA)	29.6	29.3	25.6
2 (MPE-HLDA)	29.0	28.8	25.4

Table 1. CTS unadapted 15-frame MPE-HLDA compared to baseline ML HLDA.

The idea of using held-out data and a strong language model is to approximate the testing scenario as much as possible.

5. RESULTS

5.1. CTS Results

On the CTS corpus, we ran both unadapted and adapted experiments with frame concatenated PLP cepstra. We measured the performance of MPE-HLDA on the EARS 2003 Evaluation test set (Eval03), and in some cases on the 2004 Development test set (Dev04).

In the unadapted experiments, we used non-crossword state clusters for acoustic modeling, LDA for the first-level projection L, and HLDA for the second-level initial projection A_0. We concatenated 15 frames, hence $\ell = 225$, and we chose $n = 130$ and $p = 60$ as input and output dimensionality. Two main iterations of joint optimization of the matrix and covariance statistics were performed, using the procedure outlined in section 3.

The effect of MPE-HLDA on recognition accuracy is shown in Table 1, where we can see significant WER reductions with the 12 Gaussian-per-state (12GPS) MPE-HLDA model after each main iteration of the optimization process. In particular, there is a 1.5% absolute gain on the training data with respect to the baseline ML HLDA model, and a 1.8% absolute gain on the Eval03 test set. Note, however, that our standard ML HLDA unadapted model contains about 6 times as many Gaussians as the 12GPS MPE-HLDA model, so it is more interesting to look at the effect of MPE-HLDA on the large ML training. This is shown at the rightmost column of Table 1, where we can see that the overall gain from MPE-HLDA is reduced to 0.5% absolute.

One might argue that the gain on the 12GPS MPE-HLDA model does not come from better features, but from tuning the model to fit the MPE criterion. To answer this question, we performed two groups of tests using the optimized model and projection from iteration 1. In the first test, we took the optimized MPE-HLDA model from the first iteration of Table 1, and updated the Gaussian parameters with regular ML training. If the optimized feature projection had not selected better features at all, there should have been apparent degradations after ML updating, however, it turned out that the degradation was insignificant (Table 2). In the second test, we took both the initial and first iteration MPE-HLDA models, and updated their Gaussians under the MPE criterion. The results showed that even after MPE training of the Gaussian parameters there is a 0.7% absolute gain due to MPE-HLDA (Table 3).

As an alternative initial projection A_0, we also tried a a 60×130 MLLT matrix[2]. It turned out that it had as good performance

[2]First a 60×60 MLLT transform was estimated in the projected space defined by the first level LDA, then it was padded with zeros to make a 60×130 projection.

ML Iter.	Eval03 WER
0	29.3
7	29.4

Table 2. ML updating of optimized MPE-HLDA model (Model 1 of Table 1).

Model	Eval03 WER
0 (HLDA)	25.8
1 (MPE-HLDA)	25.1

Table 3. MPE updating of initial and optimized MPE-HLDA models (Model 0 and 1 of Table 1).

as HLDA but was less expensive to run, so we adopted it as the initialization method in later experiments.

Given the gains from MPE-HLDA, we then used it in conjunction with HLDA-SAT for estimating the global speaker independent feature projection. For best performance, crossword models were used in MPE-HLDA. After the optimal projection was found, we plugged it into HLDA-SAT to build a full system and decoded both Eval03 and Dev04. Table 4 shows the WERs during MPE-HLDA training and on testing. The results verified the improved performance due to MPE-HLDA.

Iteration	MPE-HLDA model tr. WER	ML SAT model test WER
0 (LDA+MLLT)	26.1	17.3
1 (MPE-HLDA)	25.3	16.9

Table 4. Effect of MPE-HLDA on adapted CTS ML SAT models. Test WER measured on the Eval03+Dev04 set.

5.2. BN results

Table 5 shows the WERs on training data and on h4d04 test data. The configuration of MPE-HLDA experiments was the same as with the adapted experiments on CTS, except that we also experimented with increased number of frames to concatenate.

We see two interesting results from the experiments. First, the MPE-HLDA is more robust than LDA+MLLT when the dimensionality increases. As the number of frames went from 9 to 23, LDA+MLLT performance progressively degraded, while MPE-HLDA recovered the loss even from the bad LDA+MLLT initial point. Second, we chose to use 15 and 23 frames in doing MPE-HLDA because previous experiments at BBN showed that features from 9 frames gained little over features from base cepstra and deltas. On the BN corpus, however, results show that the use of more than 9 frames does not provide any further benefit. This leads us to believe that the nature of this corpus is different from that of CTS. It is unclear what is the exact reason, but we know that there are more silences and music between the speech in the BN data, so that longer concatenation windows may suffer from including more irrelevant features.

6. CONCLUSIONS

In this paper we have taken a first look at a new feature analysis method, MPE-HLDA. Its application to both unadapted and speaker-adapted training shows that it is effective in reducing recognition errors, and that it is more robust than other commonly used analysis methods like LDA and HLDA.

We believe that MPE-HLDA can be improved further. In future work, we are planning to explore other input features besides cepstra, and to use multiple projections and more efficient modeling in the original feature space.

Frames	Projection	n	tr. WER	h4d04 WER
9	LDA+MLLT	-	12.7	12.8
9	MPE-HLDA	135	12.7	12.7
15	LDA+MLLT	130	12.7	12.9
15	MPE-HLDA	130	12.2	12.7
23	LDA+MLLT	190	13.1	13.2
23	MPE-HLDA	190	12.1	12.7

Table 5. Effect of MPE-HLDA on adapted BN ML SAT models.

7. REFERENCES

[1] G. Saon et al., "Maximum likelihood discriminant feature spaces," in *Proceedings of ICASSP*, June 2000, vol. 2, pp. II1 129–II1 132.

[2] N. Kumar and A. G. Andreou, "A generalization of linear discriminant analysis in maximum likelihood framework," Tech. Rep. JHU-CLSP Technical Report No. 16, Johns Hopkins University, Aug. 1996.

[3] M. J. F. Gales, "Maximum likelihood multiple subspace projections for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 37–47, Feb. 2002.

[4] V. Valtchev et al., "MMIE training of large vocabulary recognition systems," *Speech Communication 22*, pp. 303–314, June 1997.

[5] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of ICASSP*, 2002.

[6] X. Liu and M. J. F. Gales, "Discriminative training of multiple subspace projections for large vocabulary speech recognition," Tech. Rep. CU Technical Report No. 489, Cambridge University, England, Aug. 2004.

[7] S. Matsoukas and R. Schwartz, "Improved speaker adaptation using speaker dependent feature projections," in *Proceedings of ASRU*, Virgin Islands, U.S., Nov. 2003, pp. 273–278.

[8] R. Schwartz et al., "Speech recognition in multiple languages and domains: the 2003 BBN/LIMSI EARS system," in *Proceedings of ICASSP*, Montreal, Quebec, Canada, May 2004, vol. III, pp. 753–756.