

Log-Linear Framework for Linear Feature Transformations in Speech Recognition

Muhammad Ali Tahir, Georg Heigold, Christian Plahl, Ralf Schlüter, Hermann Ney

*Chair of Computer Science 6, Computer Science Department
RWTH Aachen University, Aachen, Germany*

{tahir, heigold, plahl, schlueter, ney}@informatik.rwth-aachen.de

Abstract—Linear Discriminant Analysis (LDA) has been established as an important means for dimension reduction and decorrelation in speech recognition. The major points of criticism of LDA are that it uses an ad hoc and non-discriminative training criterion, and that the estimation is performed in a separate preprocessing step. This paper presents a new discriminative training method for the estimation of (projecting) linear feature transforms. More precisely, the problem is formulated in the log-linear framework, resulting in a convex optimization problem. Experimental results are provided for a digit string recognition task to compare the performance and robustness of the proposed approach (in combination with ML or MMI optimized acoustic models) with conventional LDA. Also, first experiments for a large vocabulary task are presented.

I. INTRODUCTION

Linear Discriminant Analysis (LDA) is a technique to improve the feature representation for classification purpose. It results in a transformation matrix which can also be used for dimension reduction. In [1], LDA was first used as a pre-processing step for a hidden-Markov model based speech recognition system, and since then has become an integral part of a modern automatic speech recognition system. In a Maximum Likelihood (ML) framework it guarantees the best feature representation for a given number of dimensions [2]. The discriminative training approach for training Gaussian density parameters of HMM in general offers better performance than the conventional ML training. A frequently used discriminative objective function is Maximum Mutual Information (MMI) [3]. The parameters of the feature transformation matrix are calculated by LDA, but alternatively they could also be trained discriminatively. One such example is [4] where an iterative optimization is used to directly train a reduced dimension feature transformation matrix. Another one is [2]; here the transformation matrix is trained assuming unequal class covariances for Gaussian densities. An interesting work is fMPE [5], where a matrix is trained that projects a very high-dimensional posterior feature vector to the original feature dimension and then added to it. Constrained maximum likelihood linear regression (CMLLR) are feature transforms that are specifically focused on speaker-related characteristics [6].

In [7] it is shown that a Gaussian HMMs can be represented as log-linear models. The training of Log-linear models or

Conditional Random Fields (CRF) is a convex optimization problem. The global optimum is determined by Generalized Iterative Scaling (GIS). More efficient schemes have been also proposed recently e.g. RPROP [8]. For linear feature transforms, the optimization problem can be written such that the elements of the transformation matrix appear as log-linear parameters. The resulting training is termed as log-LDA.

This paper investigates the log-linear training of feature transformation matrix using the MMI criterion. The mathematical details of the training are presented. Its performance is compared to the classical LDA. Furthermore, its behavior is explored on some of the open problems related to the classical LDA such as the increasing temporal window size and strong linear dependencies between features. Effect of an additional regularization term and alternating optimization between acoustic model and transformation parameters has also been investigated.

The remainder of this paper is organized as follows. In Section II LDA is briefly described. In Section III the linear feature transformation and the conversion of gaussian HMMs to log-linear models is presented, along with the optimization procedure. Section IV contains the experimental results and their findings. Finally, the conclusions are presented in Section V.

II. CONVENTIONAL LDA

The purpose of Linear Discriminant Analysis (LDA) is to find a linear transformation that best separates two or more classes. In speech recognition we get a sequence of continuous feature vectors as a result of signal processing. Therefore we use LDA to find a weighted combination of these features so that the separation between different classes is maximized. The classes here correspond to different phoneme models. We can achieve a dimensionality reduction if we drop some of the insignificant dimensions.

Suppose we have vectors $x_1^T = (x_1, \dots, x_T)$ where $x \in \mathbf{R}^D$ corresponding to $c = 1, \dots, C$ different classes. The within class scatter of the data is Σ and the between class scatter is Σ_b . We need to find a transformation that maximizes the ratio of between class scatter to within class scatter. the transformation matrix for LDA is the matrix containing eigenvectors of $\Sigma^{-1}\Sigma_b$ as its rows. To achieve dimension

reduction, we could drop the rows corresponding to smaller eigenvalues.

III. LINEAR FEATURE TRANSFORMATIONS

In this section we explain how the Gaussian mixture models can be represented as their equivalent log-linear form. Similarly, the transformation parameters are also converted to log-linear form. The optimization criteria at two different levels i.e. frame level and sentence level are described along with their optimization procedure.

A. Log-Linear Mixture Models

Let the feature vectors x_1^T belonging to $c = 1, \dots, C$ classes, each class with Gaussian parameter set $\theta_c = \{\mu_c, \Sigma_c\}$. Then the conditional probability is

$$\begin{aligned} p_\theta(c|x) &= \frac{p(c)p_\theta(x|c)}{\sum_{c'} p(c')p_\theta(x|c')} \\ &= \frac{\exp(x^\top \Lambda_c x + \lambda_c^\top x + \alpha_c)}{\sum_{c'} \exp(x^\top \Lambda_{c'} x + \lambda_{c'}^\top x + \alpha_{c'})} \end{aligned} \quad (1)$$

The last step of the above equation contains the new parameters $\Lambda_c \in \mathbf{R}^{D \times D}$, $\lambda_c \in \mathbf{R}^D$ and $\alpha_c \in \mathbf{R}$ in log-quadratic form. This form directly models the posterior probability and the exponential term in the numerator does not represent a true probability density in x space.

If a pooled covariance matrix Σ is used, the covariance in the numerator and denominator cancels out and the resulting form is log-linear with no second order term in the exponent.

$$p_\theta(c|x) = \frac{\exp(\lambda_c^\top x + \alpha_c)}{\sum_{c'} \exp(\lambda_{c'}^\top x + \alpha_{c'})} \quad (2)$$

for Gaussian mixtures

$$p_\theta(c|x) = \frac{\sum_l \exp(\lambda_{c,l}^\top x + \alpha_{c,l})}{\sum_{c',l} \exp(\lambda_{c',l}^\top x + \alpha_{c',l})} \quad (3)$$

for $l = 1 \dots L_s$ mixture parameters in each class c .

B. Linear Feature Transforms: Log-Linear Representation

A transformation matrix $A \in \mathbf{R}^{D' \times D}$ that transforms features as $y = Ax$ can be included into Equation(2)

$$\begin{aligned} p_{\Lambda,A}(c|x) &= \frac{\exp(\lambda_c^\top Ax + \alpha_c)}{\sum_{c'} \exp(\lambda_{c'}^\top Ax + \alpha_{c'})} \\ &= \frac{\exp\left(\sum_{d',d} a_{d',d} (\lambda_{c,d'} x_d) + \alpha_c\right)}{\sum_{c'} \exp\left(\sum_{d',d} a_{d',d} (\lambda_{c',d'} x_d) + \alpha_{c'}\right)} \end{aligned} \quad (4)$$

A is a projective transformation. λ_c and α_c are as in Equation(2). $\lambda_{c,d}$ is the d^{th} scalar component of vector λ_c , x_d is the d^{th} component of x , and $a_{d',d}$ is the element of matrix A at d'^{th} row and d^{th} column. If the class parameters $\lambda_{c,d}$ are held constant, then the equation is log-linear with respect to parameters $a_{d',d}$.

The MMI optimization $\mathcal{F}(\Lambda, A) = \sum_{n=1}^N \log p_{\Lambda,A}(c_n|x_n)$ can be done either at frame level ($c = s$) or sentence level

($c = (s_1^T, W)$). For an introduction to frame or sentence level optimization, refer to [9]

C. MMI on Frame Level

The frame level objective function is

$$\begin{aligned} \mathcal{F}^{(frame)}(\Lambda, A) &= -\tau_A \|A\|^2 - \tau_\Lambda \|\Lambda\|^2 \\ &\quad + \sum_{r=1}^R \sum_{t=1}^{T_r} w_s \log p_{\Lambda,A}(s_t|x_t) \end{aligned} \quad (5)$$

$$p_{\Lambda,A}(s_t|x_t) = \frac{\exp\left(\lambda_{s_t}^\top Ax_t + \hat{\alpha}_{s_t}\right)}{\sum_{s'} \exp\left(\lambda_{s'}^\top Ax_t + \hat{\alpha}_{s'}\right)} \quad (6)$$

for a fixed alignment s_1^T , transformation matrix A and state parameters $\Lambda_s = \{\lambda_s, \alpha_s\}$. τ_A and τ_Λ are regularization parameters. w_s are state weights which can be adjusted to give e.g. less weight to noise and silence states. $\hat{\alpha}_s = \alpha_s + \log p(s)$, $p(s)$ is the prior probability of state s and R is the total number of sentences in the training corpus. The state priors $p(s)$ are kept fixed during log-LDA training. These are added to α_s for training, and then later subtracted for recognition. The frame level criterion is similar to classical LDA because of its class definition.

D. MMI on Sentence Level

The sentence level MMI objective function is

$$\begin{aligned} \mathcal{F}^{(sentence)}(\Lambda, A) &= -\tau_A \|A\|^2 - \tau_\Lambda \|\Lambda\|^2 \\ &\quad + \sum_{r=1}^R \log \frac{\left(p(W_r) p_{\Lambda,A}(X_r|W_r) \exp(\rho \delta(W_r, W_r))\right)^\gamma}{\sum_{W \in \mathcal{M}_r} \left(p(W) p_{\Lambda,A}(X_r|W) \exp(\rho \delta(W, W_r))\right)^\gamma} \end{aligned} \quad (7)$$

$$p_{\Lambda,A}(X_r|W) = \sum_{s_1^T | W} \left\{ \prod_{t=1}^{T_r} p(s_t | s_{t-1}) \exp\left(\lambda_{s_t}^\top Ax_t + \alpha_{s_t}\right) \right\} \quad (8)$$

where \mathcal{M}_r is the set of all possible word sequences, ρ is a scaling factor for the margin term and γ is scaling for the posterior term. In the numerator, only the best state sequence is used. In the denominator the summation space does not change if exact denominator is used i.e. all possible word sequences are evaluated. However, in our experiments the lattice approximation is used for word sequences [10].

However, it should be mentioned that for mixtures the objective function does not remain strictly log-linear, and therefore a global maximum can only be guaranteed if a single parameter λ_s is used as state emission probability.

E. Optimization

MMI optimization can be done using the GIS algorithm [11], but for this case it is found to be slower than the general purpose RPROP algorithm [8]. RPROP is a first order optimization algorithm that takes only the sign of the

partial derivatives into account. The weights for parameters are increased if there was no sign change in the partial derivatives in the last iteration, and vice versa.

For a fixed set of log-linear mixture parameters in Equation(5) and Equation(6), there is a single set of log-LDA parameters that represents the global maximum of MMI function, according to the maximum entropy principle [11]. The same is true for fixed log-LDA parameters and variable mixture parameters. A question that arises during simultaneous training of both parameter sets: It is possible to achieve the global maximum with such a procedure? There are some special cases which seem to contradict this hypothesis:

- Suppose that a particular pair of LDA matrix A and state parameters Λ is calculated using maximum likelihood training (Λ obtained by conversion from Gaussian densities θ). Now if two rows of matrix A are interchanged to create A' , then the mixture sets Λ' trained using this new matrix will also have the corresponding two elements of each log-linear parameter λ'_s interchanged. These two different setups will have the same value of MMI function. Furthermore, both of these if used for recognition should result in the same WER, because the parameter sets are essentially the same. This indicates that there can be several equally optimal maxima of the MMI function.
- A degenerate case: If during the ML training all the rows of LDA matrix are artificially made identical to each other (e.g. by putting the value of first eigenvector into all the rows), then the mixture sets Λ trained from them will also have the same values for all the elements in each λ_s . Now if the state parameters are trained log-linearly, the elements inside each λ_s will still remain equal. Even if log-LDA is done, the rows of matrix A would remain equal to each other. Thus the alternation strategy in this case would not be successful and it would get stuck in a local maximum. This shows that for alternation training, it might not always be possible to reach the global optimum.

Since A is a reduced rank matrix, therefore the objective function which was convex in the original space may not be convex in the transformed space.

The above discussion shows that a good initial guess is important if both transformation and state parameters are trained at the same time. LDA trained matrix and its corresponding Gaussian mixture densities provide a good initial guess.

IV. EXPERIMENTS AND RESULTS

A. Speech corpora

For the performance analysis of log-LDA, two speech corpora are used; a small vocabulary task SieTill and a large vocabulary European Parliament Plenary Sessions (EPPS) task.

The SieTill speech corpus is a German digit string corpus containing eleven digits (including the pronunciation variant "zwo" for "zwei") as vocabulary. The training corpus is 11.6

hours and the test corpus is 11.7 hours. It contains separate training corpora for male and female speakers as well as combined corpora for both genders. There are 11 whole word HMMs with a total of 214 states for each gender plus 1 silence state. Gaussian mixture densities with diagonal pooled covariance matrix are used. The percentage of silence is 55%.

The EPPS is a part of 2006 TC-STAR ASR evaluation campaign. It contains plenary session speeches of the European Parliament in British English. The vocabulary size is 54k words, with a training corpus of 40.8 hours and evaluation corpus of 3.5 hours. The acoustic model is across-word using triphones. Trigram language model is used (perplexity of evaluation part: 99.0). The newer versions of this task contain more than 100 hours of training data.

In both cases, MFCC features are used as acoustic input. The LDA transformation matrix takes a temporal window of 11×12 MFCC features and reduces it to 45 dimensions.

B. Log-LDA vs. LDA

If the transformation matrix is directly trained by keeping ML-trained state parameters constant, it results in an optimization of the MMI objective function as well as word error rate. However, if the state parameters are also first trained log-linearly, then the improvement caused by log-LDA diminishes. In the case of sentence based log-LDA the improvement is still noticeable, while for frame based log-LDA it does not result in better WER

TABLE I
COMPARISON OF LDA AND LOG-LDA FOR DIFFERENT TRAINING CRITERIA AND ACOUSTIC MODELS, WORD ERROR RATES (WERS) FOR SIE TILL TEST CORPUS. MMI IS ON EITHER FRAME OR ON SENTENCE LEVEL BOTH FOR THE ACOUSTIC MODEL AND THE FEATURE TRANSFORM.

	Feature transform	Acoustic model	WER [%]	
			frame	sentence
Single Gaussians	LDA	ML	3.53	
		MMI	2.72	2.53
	MMI (log-LDA)	ML	3.51	2.84
		MMI	2.69	2.38
16 Gaussians per state	LDA	ML	1.91	
		MMI	1.63	1.49
	MMI (log-LDA)	ML	1.88	1.77
		MMI	1.60	1.49

TABLE II
COMPARISON OF LDA AND LOG-LDA FOR DIFFERENT TRAINING CRITERIA, WORD ERROR RATES (WERS) FOR EPPS TEST CORPUS. MMI IS ON FRAME OR SENTENCE LEVEL ON THE FEATURE TRANSFORM.

	Feature transform	WER [%]	
		frame	sentence
Single Gaussians	LDA	28.2	
	log-LDA	28.1	27.1
16 Gaussians per state	LDA	20.7	
	log-LDA	-	20.1

For frame level log-LDA the state alignment is kept fixed, and is obtained from a previous ML training. For the sentence level log-LDA, the training is done using word lattices. It is

observed that the objective function optimization is better if the lattices are re-trained after every few iterations, so that they remain current with respect to the state and transformation parameters.

For single λ_s per state, the regularization parameters τ_A and τ_Λ do not play a significant role. However for mixture of parameters, a proper choice for the constants keeps the WER from increasing again after it has reached a minimum value.

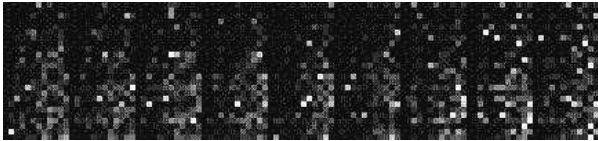


Fig. 1. Absolute of difference of the LDA and log-LDA matrix. More changes in the higher index rows (lower half of image) are observed

Another observation is that during the log-LDA training, most of the change is in the higher index rows of the transformation matrix, and there are relatively less changes in the lower index rows. This indicates that the lower index rows (that were the eigenvectors for larger eigenvalues) are already well determined from the LDA algorithm, therefore they change less during log-LDA. Fig. 1 shows the absolute of difference of LDA and log-LDA matrices shown as an intensity image. Here the black values are near zero values and white represents large positive and large negative values. This shows that the LDA matrix provides a good initial guess for initializing the log-LDA.

The results can be seen in Table I and Table II. For SieTill, a single iteration of log-LDA training requires around 2 hours, with the sentence level training requiring additional 30 minutes of lattice re-training after every few iterations. For EPPS, each iteration takes around 40 hours, and the lattice re-training takes 90 hours. These running times are with respect to one CPU, and obtained on Quad-Core AMD Opteron 2.1 GHz processor machines.

C. Robustness of Log-LDA

The robustness testing experiments described below were performed with frame level log-LDA with single acoustic parameter λ_s per state.

1) *Initialization from Scratch*: To test the convexity property of log-LDA, the state parameters λ_s were initialized from ML training by converting Gaussian model to log-linear. The transformation parameters were initialized in one case from LDA matrix, and in the other case from scratch (random numbers in the range $-10^{-5} < a_{adv} < 10^{-5}$). For the LDA initialization case the starting WER is 3.53% and after 100 iterations it reaches 3.51%. For initialization from scratch, the starting WER is 90.24% and after 100 iterations it also reaches 3.51%. This shows that for the case of single acoustic model parameter λ_s per state, the global maximum can be reached. The objective function behavior comparison between both cases can be seen in Fig. 2.

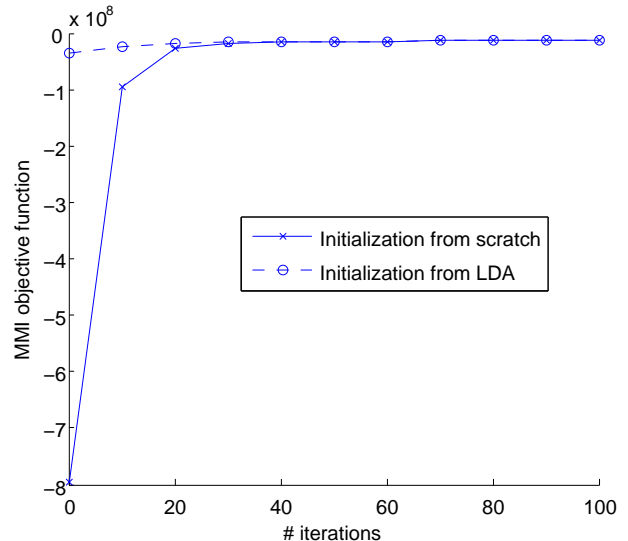


Fig. 2. SieTill, Frame level log-LDA: Comparison between initialization from scratch and initialization from classical LDA

2) *Effect of increasing window size*: In [12], experiments were performed by varying the number of consecutive MFCC frames given as input to LDA. One would expect that this would result in converging improvements, because LDA is able to filter the relevant information out of the total information given in feature vectors; therefore increasing the input window size too much should not degrade the word error rate. However, it was observed that there is a clear optimum value for the window size, after which the WER starts to increase again. During the course of this work, similar experiments were performed for the SieTill corpus, to see the effect of window size on that task. This resulted in quite similar results as in [12].

The purpose of these experiments is to see whether log-LDA can remedy this dependence of LDA on feature vector window length. Tests were performed first with normal LDA on window sizes of 5, 11 and 21, for a single λ_s per state. The optimum window size is 11, as seen in Table III; Using log-LDA decreases the WER in case of larger window size, but it does not seem to result in converging improvements.

TABLE III
SIE TILL: WER(%) FOR DIFFERENT TEMPORAL WINDOW SIZES

Window size	5	11	21
LDA	3.21	2.72	3.13
log-LDA	3.21	2.69	3.06

One possible reason for the non-converging behavior of LDA and log-LDA could be the linear dependencies between consecutive frames. For the window size of 21, the eigenvalue verification tolerance of the generalized eigenvalue algorithm in LDA needed to be increased from 10^{13} to 10^{19} , because eigenvalue checking step failed for the initial value. This shows a near singularity for window size 21, and could result in ill-determined eigenvalues and eigenvectors.

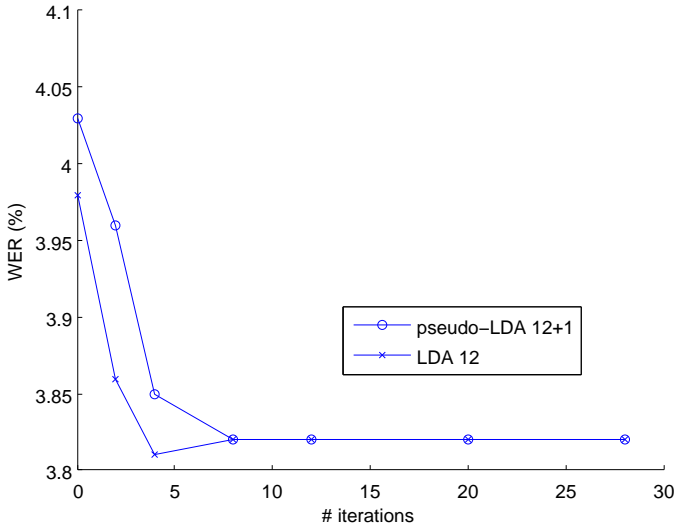


Fig. 3. SieTill: WER vs. no. of iterations for frame level log-LDA optimization of MFCC features to compare 12 and 12 + 1 coefficients

3) *Dependency of Features*: Linear dependencies of features may affect the performance of conventional LDA, as pointed out in [13]. To test this issue for log-LDA we carried out a toy experiment. The first MFCC coefficient (energy value) is duplicated twice, which causes a singularity in the LDA algorithm. This is because now the scatter matrices do not have full rank, and there should be a zero eigenvalue. As expected, this causes a degradation in the word error rate. The experiment was done on the male part of SieTill corpus (both for training and recognition). Using normal LDA with a window size of 5 frames, WER for 12 MFCC coefficients is 3.98%, but when the first coefficient (energy value) is duplicated to make 13 coefficients, the WER becomes 5.15%. There is a general case of LDA called Pseudo-LDA, that handles the problem of linear dependency between features. Pseudo-LDA differs from regular LDA in only that here the pseudo-inverse replaces the normal inverse for calculating Σ^{-1} . Using pseudo-LDA gives a WER of 4.04%, which is significantly better than the one obtained by LDA, and is comparable to the one for 12 MFCC coefficients. For the 12 coefficient case, using log-LDA (frame based) after LDA reduces the WER to 3.82%. For 12 + 1 coefficients, log-LDA after pseudo-LDA reaches 3.82%. This can be seen graphically from Fig. 3. Therefore it appears that unlike LDA, the log-LDA is not affected by strong-linear dependencies.

D. Integration of Log-LDA into Acoustic Model Training

For better optimization of MMI function, it could be useful to alternate the optimization of log-linear acoustic model parameters and log-LDA. To test this, experiments were performed by first doing some iterations of log-LDA. Then the newly calculated transformation matrix and the maximum likelihood trained single densities were given as input to the log-linear mixture training algorithm. After that, the new log-linear parameters for states were kept constant and log-LDA

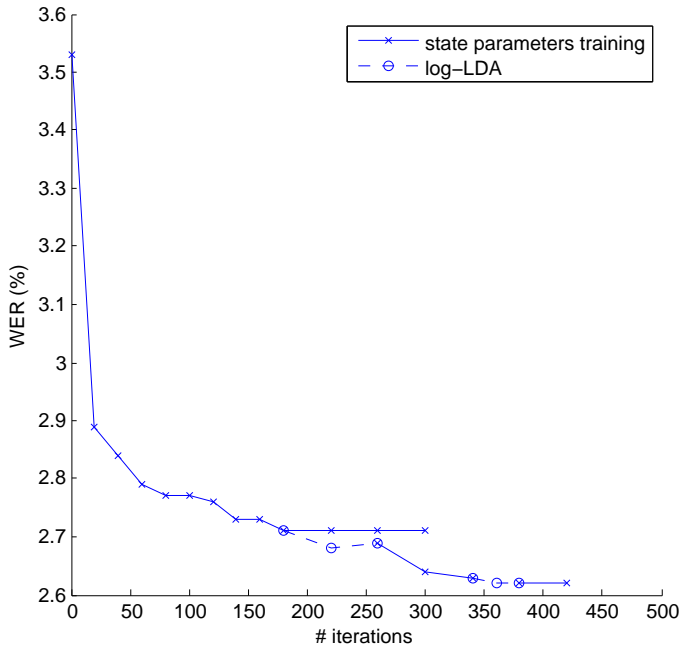


Fig. 4. SieTill: WER vs. no. of iterations for alternation between log-linear acoustic model and log-LDA training (frame level MMI)

matrix was again trained. The log-LDA was trained 2 times and state parameters 3 times with alternation. As seen from the results, this has resulted in good word error rate improvements. Although most of the improvement comes for state parameter training, still the log-LDA helps to adjust the transformation matrix to the newly calculated state parameters. If only the state parameters are trained individually, this does not result in the same level of improvement as with alternation. This is evident from Fig. 4, where the training sequence is plotted with respect to number of iterations.

V. CONCLUSIONS

We proposed a new training algorithm for (projecting) linear feature transforms. The training algorithm uses a convex training criterion such that LDA, or another reasonable estimate of the feature transform were not needed for initialization. Experimental results on the digit string recognition task SieTill were shown to compare this approach with conventional LDA. The proposed training method was used both as a preprocessing step similar to LDA and as a postprocessing step similar to fMPE. Small but consistent improvements over conventional LDA were observed for the ML optimized acoustic model while there was no significant benefit for the MMI optimized acoustic model. In addition, robustness issues (e.g. length of temporal context, feature dependencies) were investigated. The proposed approach tends to be more robust than LDA. Finally, first experimental results were shown for a large vocabulary continuous speech recognition task. Testing the convex sentence-based training criterion in [10] and evaluating the utility of the refined approach for more challenging setups, will be the next steps.

ACKNOWLEDGEMENTS

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

REFERENCES

- [1] R. Häb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proc. ICASSP'92*, vol. 1, San Francisco, USA, Mar. 1992, pp. 13–16.
- [2] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *SPECOM*, vol. 26, no. 4, pp. 283–297, Dec. 1998.
- [3] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge, England, 2004.
- [4] M. K. Omar and M. Hasegawa-Johnson, "Maximum conditional mutual information projection for speech recognition," in *Proc. INTERSPEECH'03*, Geneva, Switzerland, Sep. 2003.
- [5] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP'05*, Philadelphia, USA, May 2005.
- [6] M. Ferras, C. Leung, C. Barras, , and J. Gauvain, "Constrained MLLR for speaker recognition," in *Proc. ICASSP'07*, Honolulu, Hawaii, Apr. 2007.
- [7] G. Heigold, P. Lehnen, R. Schlüter, and H. Ney, "On the equivalence of Gaussian and log-linear HMMs," in *Proc. INTERSPEECH'08*, Brisbane, Australia, Sep. 2008.
- [8] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. ICNN'93*, San Francisco, USA, 1993, pp. 586–591.
- [9] D. Kershaw, T. Robinson, and M. Hochberg, "Context-dependent classes in a hybrid recurrent network-HMM speech recognition system," in *Proc. NIPS'96*, Denver, USA, Nov. 1996, pp. 750–756.
- [10] G. Heigold, D. Rybach, R. Schlüter, and H. Ney, "Investigations on convex optimization using log-linear HMMs for digit string recognition," in *Proc. INTERSPEECH'09*, Brighton, U.K., Sep. 2009.
- [11] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Annals of Mathematical Statistics*, vol. 43, pp. 1470–1480, 1972.
- [12] L. Welling, "Merkmalsextraktion in spracherkennungssystemen für grossen wortschatz," Ph.D. dissertation, RWTH Aachen, Aachen, Germany, 1999.
- [13] R. Schlüter, A. Zolnay, and H. Ney, "Feature combination using linear discriminant analysis and its pitfalls," in *Proc. INTERSPEECH'06*, Sep. 2006, pp. 345–348.