EXTENDED LINEAR DISCRIMINANT ANALYSIS (ELDA) FOR SPEECH RECOGNITION

G. Ruske, R. Faltlhauser, T. Pfau

Institute for Human-Machine-Communication, Technical University of Munich, Arcisstr. 21, 80290 München
Tel.: +49 89 289-28563, Fax: +49 89 289-28535, e-mail: {rus,fal,pfa}@mmk.e-technik.tu-muenchen.de

ABSTRACT

Speech recognition systems based on hidden Markov models (HMM) favourably apply a linear discriminant analysis transform (LDA) in order to get low-dimensional and uncorrelated feature components. However, since the distributions in the HMM states usually are modeled by mixture gaussian densities, the description by second-order moments (scatter matrices) no longer is correct. For this purpose we introduced a new „extended linear discriminant analysis“ transform (ELDA) which starts from conventional LDA. The ELDA transform is derived by use of a gradient descent optimization procedure based on a „minimum classification error“ (MCE) principle, which is applied to the original high-dimensional pattern space. The transform matrix, the best fitting prototype of the correct class (i.e. HMM state) and the nearest rival are adapted. We developed a method which additionally updates all prototypes by a separate maximum likelihood (ML) estimation step. This avoids that such means and covariances, which mostly remain unaffected by the MCE procedure, may diverge step by step.

1. INTRODUCTION

Automatic speech recognition systems commonly are separating this task into several subtasks, which perform the preprocessing of the speech signal, extract suitable features and finally carry out classification by means of stochastic models, such as hidden Markov models (HMM). It is obvious, that an optimal classification scheme needs all these three stages to be performed as good as possible with respect to the overall error rate. For preprocessing usually smoothed frequency spectra or cepstral coefficients along a mel scale are used with good success. In our system, we apply a functional model of human loudness sensation which yields so-called loudness spectra arranged on a Bark scale [1]. These loudness spectra have also proven to be well suited as a basis for speech recognition.

The remaining critical stage is the feature extraction, which should deliver components having special properties:

- the components should be uncorrelated since usually the HMM classifier only applies diagonal covariance matrices,
- the resulting number of components (dimensionality) should be as small as possible in order to reduce the number of parameters to be estimated during training,
- and last not least the low-dimensional feature representation should guarantee minimal classification error.

The well-known linear discriminant analysis (LDA) has proven to be a good starting point for feature extraction. In [2] this transform is further adapted with respect to minimizing the classification error using the steepest descent method. However, this additional „step transform“ was applied to the low-dimensional feature space. Several methods have been proposed to go further and to combine optimization of feature extraction with optimization of the classifier parameters, e.g. for hidden Markov models [3,4,5,6] or for simple nearest neighbor classifiers [7]. The transforms also were represented by neural nets which are discriminatively trained [8]. In contrast to most of these approaches, in our system we consequently utilize the important potential of the feature extraction stage, that is the possible reduction of dimensionality. A comparable solution already was reported in [3] for spoken letters. In our approach we are able to derive from a high-dimensional raw pattern vector or even from a series of consecutive pattern vectors (so-called super vectors) the desired low-dimensional feature vectors, which facilitate the estimation of the remaining classifier parameters considerably.

For this purpose we introduced a discriminant transform which also is derived from a linear discriminant analysis. We call this procedure „extended linear discriminant analysis“ (ELDA). The basic idea for ELDA is to start from a conventional LDA including dimensionality reduction, but to optimize this rectangular transform matrix further with respect to minimizing the overall error rate. This is necessary, since the LDA only utilizes the between-class scatter matrix and the within-class scatter matrix. Therefore, optimality is only guaranteed for the case that each class is represented by a single gaussian distribution. But this assumption is extremely violated in case of continuous density HMMs, which incorporate mixture densities consisting of several gaussian densities in each state. When the states are assumed to be discriminated by the feature extraction step, the necessary assumptions are in no way fulfilled.

In order to reach this goal, the LDA matrix is further processed in our new approach by using a gradient descent optimization procedure on the basis of a minimum classification error (MCE) principle [3], which is applied now to the original high-dimensional pattern space. The optimal solution is found by simultaneously changing both, the transform matrix and the main classifier (HMM) parameters. We start from labeled speech training material, where the alignment to the states of phoneme HMMs is known. The states of all HMMs are the classes to be discriminated by the LDA or ELDA. The error function takes into account for each input vector the correct state (correct class) as well as that state from all remaining states, which fits best (this is a „rival“ state which is assumed

to contain most danger of possible misclassification). Thus, only the parameters (gaussian distributions) of the best modes in these two selected states are changed. The correct state is always positively adapted to the input vector and the rival state is adapted away from the input vector. In this way the discriminative power of the HMMs is increased. Additionally - as the main scope of this paper - the parameters of the transform matrix are adjusted accordingly, thus increasing the discriminative power of the transform.

The main problem is based on the fact that on the one hand the total transform matrix always is adapted, but on the other hand only the selected means and covariances are touched. Therefore the remaining unaffected means and covariances will diverge step by step, until they are likewise selected as nearest prototypes. If this does not happen often enough, most of the prototypes will no longer be representative in this new feature space and the complete procedure is predicted to diverge and to produce bad overall results.

We propose two solutions to overcome this problem:

1) all prototypes (means and covariance matrices) are subjected to the transform, or
2) all prototypes are updated by a separate maximum likelihood (ML) estimation step.

The first solution is difficult to realize in practice, since the original vectors of the means (in the original NxN space) do not exist. In this paper we therefore favor the second solution. That means that an additional ML estimation step will adjust all prototypes, regardless if they are selected by the gradient adjustments or not. Of course the means will loose their special power now for discriminating the classes. But since the primary goal is to establish the ELDA transform, this goal will be reached nevertheless. The means are only to be seen here as intermediate auxiliary prototypes, which are utilized as substitutes for the subsequent HMM models which are built up in the training phase.

The methods for establishing the conventional LDA are assumed to be well-known. In the following we present the complete algorithm for ELDA: i.e. we define the error measure for discriminating HMM states and give the formulae for calculating the gradients with respect to all parameters.

2. EXTENDED LINEAR DISCRIMINANT ANALYSIS (ELDA)

We start from the average intra-class scatter matrix $\underline{S}_w$ („within classes“) and the average inter-class scatter matrix $\underline{S}_b$ („between classes“), where the classes are chosen to be the HMM states. The matrix $\underline{S}_w$ is calculated from all input vectors $\underline{x}$. A 2-step procedure creates the LDA matrix $\underline{W}$, as usual [9].

The transform matrix $\underline{W}$ is now adapted further with respect to the minimum classification error. The classifier for ELDA optimization is a simplified version of the phoneme HMMs. It consists of the states alone, whereby only the best fitting mode in a state is taken into account. Therefore, the emission probability of a pattern vector $\underline{y}$ in state i with l = 1...L modes is approximated by:

$$p(\underline{y}\,|\,s=i) \approx \max_l \left\{ mix_{il} \frac{1}{\sqrt{(2\pi)^R \left|\underline{C}_{il}\right|}} \cdot \exp\left(-\tfrac{1}{2}(\underline{y}-\underline{m}_{il})' \underline{C}_{il}^{-1} (\underline{y}-\underline{m}_{il})\right) \right\}$$

where mix_{il} is the mixture coefficient for mode l in state i, and R is the reduced dimensionality. If the neg-log probabilities are used, we get a measure for the distance to the class means:

$$D_i(\underline{y}) = \min_l \left\{ -2 \ln mix_{il} + \ln\left|\underline{C}_{il}\right| + (\underline{y}-\underline{m}_{il})' \underline{C}_{ik}^{-1} (\underline{y}-\underline{m}_{il}) \right\}$$

A discriminant measure is defined as the difference $d(\underline{y})$ between the distance of $\underline{y}$ to the correct state $s = s_c$ and the nearest rival state $s = s_k$:

$$d(\underline{y}) = D_c(\underline{y}) - D_k(\underline{y})$$
$$= 2 \ln \frac{mix_{kl'}}{mix_{cl}} + \ln \frac{\left|\underline{C}_{cl}\right|}{\left|\underline{C}_{kl'}\right|} + (\underline{y}-\underline{m}_{cl})' \underline{C}_{cl}^{-1} (\underline{y}-\underline{m}_{cl}) - (\underline{y}-\underline{m}_{kl'})' \underline{C}_{kl'}^{-1} (\underline{y}-\underline{m}_{kl'})$$

where the letter l denotes the index for the best fitting gaussian distribution in the correct state c, and l′ denotes the best fitting distribution in the rival state k. We decided to search for the rival only in the non-correct states, i.e. $k \neq c$. Here we have some similarity to the solution reported in [7], where the mean vectors constitute a simple nearest neighbor classifier and the variances are not taken into account.

Since the LDA transform is used for dimensionality reduction, too, only R components are calculated from the original N components. That means, only a rectangular part N x R of the original N x N matrix is used. If we denote - for sake of simplicity - the pattern vector after mean-subtraction again as $\underline{x}$, we get:

$$\underline{y} = \underline{W}' \underline{x} \quad \text{with LDA matrix } \underline{W}_{N \times R} \text{ and } R < N, \quad E\{\underline{x}\} = \underline{0}$$

In the following, the covariance matrices are always assumed to be diagonal, so that matrix inversion gets trivial and only the variances C_{ii} are taken into account. Applying the transform to the actual mean-subtracted input vector $\underline{x}$ and performing some basic calculations, the distance $d(\underline{y})$ is expressed on the basis of the single components as:

$$d(\underline{y}) = 2 \ln \frac{mix_{kl'}}{mix_{cl}} + \ln \prod_{j=1}^{R} \frac{C_{jj\,l}^{(c)}}{C_{jj\,l'}^{(k)}} +$$
$$\sum_{j=1}^{R} \left[\begin{array}{l} (m_{j\,l}^{(c)})^2 (C_{jj\,l}^{(c)})^{-1} - (m_{j\,l'}^{(k)})^2 (C_{jj\,l'}^{(k)})^{-1} + \\ \left(\sum_{i=1}^{N} W_{ij}\, x_i \right)^2 \left((C_{jj\,l}^{(c)})^{-1} - (C_{jj\,l'}^{(k)})^{-1} \right) + \\ 2 \left(\sum_{i=1}^{N} W_{ij}\, x_i \right) \left(\begin{array}{l} m_{j\,l'}^{(k)} (C_{jj\,l'}^{(k)})^{-1} \\ - m_{j\,l}^{(c)} (C_{jj\,l}^{(c)})^{-1} \end{array} \right) \end{array} \right]$$

If the distance measure is less than zero, we have a correct classification, because the rival state k is farther away than the

correct state c. On the contrary, a $d(\underline{y})$ value greater than zero denotes a wrong classification.

Using the sigmoid function, a loss function is defined as:

$$L(\underline{y}) = f(d(\underline{y})) = \frac{1}{1 + e^{-\gamma\, d(\underline{y})}}$$

with the derivative

$$\frac{\partial L(\underline{y})}{\partial\, d(\underline{y})} = \frac{\gamma\, e^{-\gamma\, d(\underline{y})}}{(1 + e^{-\gamma\, d(\underline{y})})^2} = \gamma\, f(d(\underline{y}))\ (1 - f(d(\underline{y})))$$

The total loss L for all vectors $\underline{y}_m$, m = 1...M is calculated as the sum

$$L = \sum_{m=1}^{M} L(\underline{y}_m)$$

The loss function gets a small value, if the distances $d(\underline{y}_m)$ become as negative as possible, and that means that we have correct classifications. A (local) minimum for L is obtained by using a gradient descent method. The parameters of an arbitrary function Φ have to be adjusted in an iterative manner from step n to step n+1 with the adaptation constant ε:

$$\Phi(n+1) = \Phi(n) - \varepsilon \frac{\partial L}{\partial\, \Phi(n)} \quad \text{with } \varepsilon = \text{adaptation constant}$$

In this way the LDA matrix $\underline{W}$, the two selected mean vectors $\underline{m}^{(c)}_{l}$ and $\underline{m}^{(k)}_{l'}$, and the corresponding diagonal covariance matrices can be recalculated for each component:

$$W_{ij}(n+1) = W_{ij}(n) - \varepsilon_1 \frac{\partial L}{\partial\, W_{ij}(n)}$$

$$m^{(c)}_{i\,l}(n+1) = m^{(c)}_{i\,l}(n) - \varepsilon_2 \frac{\partial L}{\partial\, m^{(c)}_{i\,l}(n)}$$

$$m^{(k)}_{i\,l'}(n+1) = m^{(k)}_{i\,l'}(n) - \varepsilon_3 \frac{\partial L}{\partial\, m^{(k)}_{i\,l'}(n)}$$

$$C^{(c)}_{ii\,l}(n+1) = C^{(c)}_{ii\,l}(n) - \varepsilon_4 \frac{\partial L}{\partial\, C^{(c)}_{ii\,l}(n)}$$

$$C^{(k)}_{ii\,l'}(n+1) = C^{(k)}_{ii\,l'}(n) - \varepsilon_5 \frac{\partial L}{\partial\, C^{(k)}_{ii\,l'}(n)}$$

Using the chain rule we get the desired gradients for one input vector $\underline{x}$ (see the box of equations below).

These gradients have to be summed up for all input vectors, in order to get an adaptation „all at a time“. For instance, the gradients for the total loss with respect to the $\underline{W}$ matrix component W_{ij} are obtained by

$$\frac{\partial L}{\partial W_{ij}} = \sum_{m=1}^{M} \frac{\partial L(\underline{y}_m)}{\partial W_{ij}}$$

The adaptation constants $\varepsilon_{1...5}$ have to be well adjusted in order to avoid corrupted results. For instance, the variances should be updated with a small value, since they could get too narrow or even get negative.

It has to be noticed, that the gradients of the means and covariance matrices are present only for those instances, which are selected as „correct“ or „rival“ at the moment. That means, many of the prototypes mostly remain untouched. For this reason the number uf update contributions may be rather different for the single items. This has to be taken into account when setting the adaptation constants.

The big computational load appears only in the training phase. During application, we can apply the transform matrix $\underline{W}$ as usual, reducing the dimensionality of the input vectors from N to R. Since we start from a classical LDA, the results are guaranteed to be at least as good as those obtained with LDA. However, we can expect that the new ELDA matrix obtained

$$\frac{\partial L(\underline{y})}{\partial\, W_{ij}} = 2\,\gamma\, f(d(\underline{y}))\,(1 - f(d(\underline{y}))) \cdot \left(\left((C^{(c)}_{jj\,l})^{-1} - (C^{(k)}_{jj\,l'})^{-1} \right) \left(\sum_{n=1}^{N} W_{nj}\, x_n \right) x_i + \left(m^{(k)}_{j\,l'} (C^{(k)}_{jj\,l'})^{-1} - m^{(c)}_{j\,l} (C^{(c)}_{jj\,l})^{-1} \right) x_i \right)$$

$$\frac{\partial L(\underline{y})}{\partial\, m^{(c)}_{j\,l}} = 2\,\gamma\, f(d(\underline{y}))\,(1 - f(d(\underline{y}))) \cdot \left(\left(m^{(c)}_{j\,l} - \left(\sum_{n=1}^{N} W_{nj}\, x_n \right) \right) (C^{(c)}_{jj\,l})^{-1} \right)$$

$$\frac{\partial L(\underline{y})}{\partial\, m^{(k)}_{j\,l'}} = 2\,\gamma\, f(d(\underline{y}))\,(1 - f(d(\underline{y}))) \cdot \left(\left(-m^{(k)}_{j\,l'} + \left(\sum_{n=1}^{N} W_{nj}\, x_n \right) \right) (C^{(k)}_{jj\,l'})^{-1} \right)$$

$$\frac{\partial L(\underline{y})}{\partial\, C^{(c)}_{jj\,l}} = \gamma\, f(d(\underline{y}))\,(1 - f(d(\underline{y}))) \cdot \left((C^{(c)}_{jj\,l})^{-1} - \left((m^{(c)}_{j\,l})^2 + \left(\sum_{n=1}^{N} W_{nj}\, x_n \right)^2 - 2 \left(\sum_{n=1}^{N} W_{nj}\, x_n \right) m^{(c)}_{j\,l} \right) (C^{(c)}_{jj\,l})^{-2} \right)$$

$$\frac{\partial L(\underline{y})}{\partial\, C^{(k)}_{jj\,l'}} = \gamma\, f(d(\underline{y}))\,(1 - f(d(\underline{y}))) \cdot \left(-(C^{(k)}_{jj\,l'})^{-1} + \left((m^{(k)}_{j\,l'})^2 + \left(\sum_{n=1}^{N} W_{nj}\, x_n \right)^2 - 2 \left(\sum_{n=1}^{N} W_{nj}\, x_n \right) m^{(k)}_{j\,l'} \right) (C^{(k)}_{jj\,l'})^{-2} \right)$$

by the additional gradient adaptation will decrease the error rate further.

3. EXPERIMENTAL RESULTS

The methods have been tested with speech material from the German Verbmobil project. The training set consisted of 1186 sentences from 53 different speakers, spoken in a spontaneous manner. The independent test consisted of 286 sentences from 12 new speakers. The system used 52 context-independent phoneme models (HMMs) with 3...4 states. Altogether there were 169 states giving 169 classes for the LDA. The lexicon contained 3312 German words.

The preprocessing step yields loudness spectra with 20 components, delta spectra and delta-delta spectra with 20 components each. Together with the loudness values, a modified loudness component and the zero crossing rate, we get 66 components. Three consecutive vectors are collected to represent a super vector, denoted as $\underline{x}$ with 198 dimensions. This input vector is reduced by the LDA or ELDA to a 66-dimensional vector $\underline{y}$ which is the input to the HMM classifier. The sigmoid parameter γ was set to 0.5 .

The following table shows the word error rate after the individual iterations (the complete training material applied „all at a time“). The recognition test has been carried out with the independent test set.

orig. LDA	ELDA (iteration # 1)	ELDA (iteration # 2)
41.5%	40.9%	43.0%

Table 1. Total word error rate, ELDA adaptation alone.

The results in Table 1 show that the error rate slightly goes down after 1 full iteration, and then distinctly increases. This effect was expected, since the ELDA transformed feature space no longer fits to all prototypes.

In order to overcome this problem, we applied a maximum likelihood estimation step after the ELDA iteration, reducing the error rate considerably, see Table 2. A further ELDA iteration (# 4, not contained in the table) strongly rised the error rate again, since in this case we obviously already got some kind of overadaptation. Applying two ML steps gave good and robust HMM models, but the word error rate increased again. Best results were obtained by using one ELDA iteration followed by one ML step.

Orig. LDA	ELDA (iter. # 1)	ML (iter. # 2)	ML (iter. # 3)
41,5%	40,9%	40,1%	40,9%

Table 2. Total word error rate after additional ML estimation.

Altogether, with application of ELDA an error rate reduction of 1.4 % (or relatively 3.4 %) as compared to the original LDA transform could be obtained in these first experiments.

4. DISCUSSION

The variances are very sensitive to the gradient adaptation method. For this reason, in an additional experiment we disclaimed their adaptation. The results obtained were very similar to those displayed above in Table 2. That means that the main effects are caused by the ELDA transform and by adaptation of the mean vectors, and adaptation of the variances could be neglected.

Some re-classification tests with the training material itself showed, that the error rate could be reduced considerably further, down to 23 %. But in this case we already got a clear overadaptation effect. That means that the training set has to be enlarged, in order to get better results for independent test sets. It will be especially interesting to test the second method, that is to apply during the iteration steps the newly calculated ELDA transform to all means immediately, thus avoiding the mismatch of the unaffected mean vectors. These experiments are in progress. In any case, the presented methods for calculating the ELDA transform have proven to yield a suitable, improved feature extraction stage for speech recognition.

5. REFERENCES

[1] Ruske, G. and Beham, M.: Gehörbezogene automatische Spracherkennung. In: "Sprachliche Mensch-Maschine-Kommunikation", (H. Mangold, Hrsg.). Oldenbourg-Verlag, München Wien, 1992, 33 - 47.

[2] Ayer, C.M, Hunt, M.J., and Brookes, D.M.: A discriminatively derived linear transform for improved speech recognition. Eurospeech 1993, Berlin, Vol. 1, p. 583 - 586.

[3] Euler, S.: Integrated optimization of feature transformation for speech recognition. Eurospeech 1995, Madrid, Vol. 1, p. 109 - 112.

[4] Rathinavelu, C. and Deng, L.: HMM-based speech recognition using state-dependent, linear transforms on mel-warped DFT features. IEEE ICASSP 1996, Atlanta, 9 - 12.

[5] Rathinavelu, C. and Deng, L.: HMM-based speech recognition using state-dependent, discriminatively derived transforms on mel-warped DFT features. IEEE Trans. on Speech and Audio Proc., Vol. 5 No. 3, 1997, 243-256.

[6] Biem, A. and Katagiri, S.: Feature extraction based on minimum classification error/generalized probabalistic descent method. IEEE ICASSP 1993, Minn., II, 275 - 278.

[7] Paliwal, K.K., Bacchiani, M., and Sagisaka, Y.: Simultaneous design of feature extractor and pattern classifier using the minimum error training algorithm. Proc. of the IEEE Workshop on Neural Networks for Signal Processing, Boston 1995, p. 67 - 76.

[8] Rahim, M. G. and Lee, C.-H.: Simultaneous ANN feature and HMM recognizer design using string-based minimum classification error (MCE) training. Int. Conf. on Spoken Language Processing1996, Philadelpia, p. 1824 - 1827.

[9] Ruske, G.: Automatische Spracherkennung. Zweite, erweiterte Auflage, Oldenbourg-Verlag, München Wien, 1994.

This work was funded by the BMBF within the framework of the *Verbmobil*-Project.