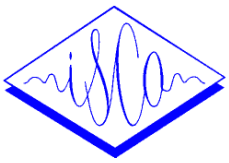

4th European Conference on
Speech Communication and Technology
EUROSPEECH '95
Madrid, Spain, September 18-21, 1995

INTEGRATED OPTIMIZATION OF FEATURE TRANSFORMATION FOR SPEECH RECOGNITION

S. Euler
Bosch Telecom
Kleyerstr. 94, D–60277 Frankfurt
e–mail: euler@fr.bosch.de

ABSTRACT

In this paper we present an new approach for obtaining an optimal linear transformation of feature vectors. The generalized probabilistic descent method is used in order to optimize the elements of a transformation matrix with respect to a functional approximation of the recognition rate of the training data. The approach is tested in a speaker dependent recognizer for spelled names.

Keywords: speech recognition, feature transformation, discriminative training

1 Introduction

Recently, a number of new approaches for optimizing the feature extraction process in speech recognition systems have been investigated. These approaches address problems such as the number of necessary features, weighting of the individual features or choosing coefficients for the calculation of the so-called Δ-coefficients. One successful and widely used method is based on the Linear Discriminant Analysis (LDA) of feature vectors [1]. The basic idea is to apply a linear transformation to the original feature vectors $\vec{y}$ in the form

$$\vec{x} = \boldsymbol{A}\vec{y}. \tag{1}$$

The matrix $\boldsymbol{A}$ is chosen such that the discrimination of the feature vectors with respect to a given set of classes is maximized. The classes can be phonetic units, states of hidden Markov models (HMM) or the densities in a set of HMMs. The dimension of the new feature vector $\vec{x}$ can be smaller then the dimension of $\vec{y}$, therefore providing a means of implicit feature selection. By using as input vectors for the LDA combinations of several e.g. cepstral input vectors $\vec{c}$ in the form

$$\vec{y}_t = (\ldots, \vec{c}_{t-1}, \vec{c}_t, \vec{c}_{t+1}, \ldots) \tag{2}$$

the transformation can replace the use of Δ-coefficients.

One at least theoretical shortcoming of the LDA for speech recognition is, that the discrimination for a given set of classes is maximized, not the recognition performance. Therefore, alternative concepts based on integrated training procedures have been proposed recently. One possibility is to extend the Maximum-Likelihood approach in order to include the optimization of the transformation matrix $\boldsymbol{A}$ [2] [3].

As an other alternative we propose in this paper an integrated training procedure based on the generalized probabilistic descent (GPD) method. The GPD approach is based on a so-called misclassification measure derived from the utterances of the training set. The misclassification measure is an approximation of the error rate in functional form. Provided that the misclassification measure is continuous with respect to a parameter a gradient descent techniques can be used for optimization of a. The approach has been applied successfully to optimize parameters of hidden Markov models such as mean and variances of the densities [4] [5]. In this paper we extend the approach to the elements of the transformation matrix $\boldsymbol{A}$. Thereby we combine two advantages, namely discriminative models and a well integrated training of the transformation matrix.

The paper is organized as follows. In the next section we summarize the basic principles of the GPD training and give the reestimation procedure for the optimization of $\boldsymbol{A}$. Then, we describe the used database and the setup of the recognition system. Finally, results are given in section 5.

2 GPD Training

Let $p_c(X)$ denote the likelihood of an utterance X for a given word model M_c. Then, for an utterance X from class c the misclassification measure $\delta(X)$ is defined as

$$\delta(X) = \log p_c(X) - \log p_w(X), \tag{3}$$

w denoting the incorrect model with the highest likelihood. In this definition a positive value of δ corresponds to a correct classification. The definition (3) focuses on the comparison between the true model and the best wrong model. A more general form of the misclassification measure using the scores from

all models can be found in [4]. Next, the misclassification measure is used to calculate a measure ℓ that approximates the recognition rate for the given utterance. We use a sigmoid function

$$\ell(X) = \frac{1}{1 + e^{-\alpha\delta(X)}} \tag{4}$$

with the constant α that defines the slope of the sigmoid function. By means of (4) the misclassification measure δ is projected into the interval $[0, 1]$. In the training procedure the GPD algorithm is applied in order to maximize ℓ. Let a be any parameter of the word model M_j. Provided that $\ell(X)$ is differentiable with respect to a, the parameter can be adjusted according to

$$\hat{a} = a + \epsilon \cdot \frac{\partial \ell(X)}{\partial a} \tag{5}$$

and

$$\hat{a} = a + \epsilon \cdot \alpha \cdot \ell(X) \cdot (1 - \ell(X)) \cdot \frac{\partial \delta(X)}{\partial a}. \tag{6}$$

Here $\hat{a}$ is the new estimate of the parameter and ϵ is a small positive constant. In case of a safe recognition ($\ell(X) \sim 1$) or a total failure ($\ell(X) \sim 0$) the term $\ell(X) \cdot (1 - \ell(X))$ and therefore the change of a becomes small. On the other hand, $\ell(X) \cdot (1 - \ell(X))$ is maximum when $\ell(X) = 0.5$, i.e. the scores for the correct and the best wrong model are equal. Therefore, the training procedure focuses on utterances which are likely to be misclassified but hopefully can be classified correctly after proper adjustment of the model parameters.

We use tied density hidden Markov models (TDHMM) without specific state transition probabilities and state duration modeling. Each model state q_n, $n = 1, \ldots, N$, is characterized by a set of weights $b_c(l, q_n)$ for the underlying set of L Gaussian density functions $f_l(\vec{x}_t)$ with diagonal covariance matrices in the form

$$f_l(\vec{x}) = \frac{(2\pi)^{-N/2}}{\prod_i \sigma_{il}} \cdot \exp\left(-\sum_i \frac{(x_{it} - \mu_{il})^2}{2\sigma_{il}^2}\right) \tag{7}$$

μ_{il} and σ_{il}^2 denoting means and variances of the density f_l. The state specific density function $p_j(\vec{x}_t|q_n)$ is then given by

$$p_j(\vec{x}|q_n) = \sum_{l=1}^{L} b_j(l, q_n) \cdot f_l(\vec{x}) \tag{8}$$

Based on the model M_j the optimum state sequence $Q_1, \ldots, Q_T$ for an utterance X with T frames is obtained by means of the Viterbi–algorithm. Then, the likelihood is given by

$$r_j(X) = \sum_{t=1}^{T} \log p_j(\vec{x}_t|Q_t) \tag{9}$$

$$= \sum_{t=1}^{T} \log\left(\sum_{l=1}^{L} b_j(l, Q_t) \cdot f_l(\vec{x}_t)\right) \tag{10}$$

It should be noted, that in the notation $p_j(\vec{x}_t|Q_t)$ the state Q_t at time t depends on the assumed model M_j.

In order to update an element a_{ij} of the transformation matrix $\boldsymbol{A}$ the differentiation of $\delta(X)$ with respect to a_{ij} is needed. Applying the chain rule results in

$$\frac{\partial \delta(X)}{\partial a_{ij}} = \tag{11}$$

$$\sum_{t=1}^{T} \sum_{l=1}^{L} \left(\frac{b_c(l, Q_t)}{p_c(\vec{x}_t|Q_t)} - \frac{b_w(l, Q_t)}{p_w(\vec{x}_t|Q_t)}\right) \cdot \frac{\partial f_l(A\vec{y}_t)}{\partial a_{ij}}$$

with

$$\frac{\partial f_l(A\vec{y}_t)}{\partial a_{ij}} = y_{ti} \cdot \frac{\mu_{lj} - x_{tj}}{\sigma_{lj}^2} f_l(\vec{x}_t) \tag{12}$$

The equations (6), (11) and (12) define the GPD reestimation procedure for the elements a_{ij} of the transition matrix $\boldsymbol{A}$.

Using e.g. the identity matrix as initial value of $\boldsymbol{A}$ (i.e. $\vec{x} = \vec{y}$) we first train the HMM parameters via Viterbi optimization. Then, in a second step, the transformation matrix is optimized. Optionally, the other HMM parameters can also be included into the GPD training. In this approach is it very easy to implement restrictions on the elements of $\boldsymbol{A}$. For example, as described above, the calculation of the Δ-coefficients is a special form of linear transformation on the extended set of input vectors. By reestimating only the corresponding elements of $\boldsymbol{A}$ and keeping the others at 0 the procedure can be used to optimize the values of the weights.

3 Database

The approach was tested in a speaker-dependent system for the recognition of spelled names. This problem provides a challenging recognition task based on a fairly small data base thereby allowing a reasonable number of the very time consuming simulation experiments. The vocabulary consisted of the 26 German letters, the umlaute Ä, Ö, Ü and the word DOPPEL (double). A sparc workstation with built-in codec and telephone handset was used for recording the speech data. The sampling rate was 8kHz and the samples were quantized with 8bit. A total of 350 utterances with 1543 letters were available for training. In order to obtain a fairly homogeneous distribution of the letters, the training utterances included names, isolated letters and random sequences of letters. 150 names (873 letters) taken from a telephone directory were used to test the system. The telephone directory contained 1580 different names. Based on the directory we constructed a binary bigramm model that specified all allowed transitions between letters. In choosing the test names from the directory we picked out names with infrequent letters such as Q or X. Nevertheless the frequency of the individual letters in the test set was very different. The most frequent letter was E (100 utterances). On the other hand the test set included only two utterances of Q.

In all simulation experiments we used LPC–analysis as start point for the feature extraction. The speech signals were divided into frames of 256 samples with an overlap of 64 samples between consecutive frames. For each speech frame a 12th–order

LPC–analysis was performed and 12 cepstral coefficients and 12 Δcepstral coefficients were calculated. Additionally one Δenergy coefficient was included in the feature vectors. The window for the calculation of the differential features was five frames for the Δcepstral and 3 frames for the Δenergy coefficients.

In order to improve the numerical stability a principal component analysis was applied to the resulting 25-dimensional feature vectors. In this way the features are normalized to mean zero and variance one. Then the corresponding matrix $\boldsymbol{H}$ was used as initial value of $\boldsymbol{A}$.

4 Simulation Setup

The word DOPPEL was modeled by an HMM with 8 states. For all other words in the vocabulary the HMM consisted of 5 states. In the recognition mode each state was duplicated in order to provide a simple constraint on the state duration. Furthermore a single state model for silence was trained. We used strict left to right models without explicit transition probabilities. The HMMs were based on a common set of 150 Gaussian density functions with diagonal covariance matrices.

As a start point for the GPD–optimization word models were trained by use of Maximum-Likelihood techniques. With the initial models word recognition rates of R_R=93.3% and R_T=87.1% were obtained for the reference and test data, respectively. Using the binary bigramm resulted in a small increase of the recognition rate to R_T=87.4%. Starting from these models the GPD–algorithm was applied iteratively to reestimate the transformation matrix.

In the formulation of the GPD training procedure isolated word recognition is assumed. Therefore, at first the boundaries of the individual letters were obtained by applying a Viterbi segmentation to the reference utterances. In this alignment pauses were allowed between the letters. However, the frames labeled as pause were not included in the GPD update. Then for each letter the scores for all word models were calculated and used as described above. In each iteration a new Viterbi segmentation was performed in order to capture the effects of the model optimization on the calculated letter boundaries. Errors resulting from insertions and deletions are not considered in this approach. We found, however, that the vast majority of the errors are confusions. For example, in a typical recognition experiment we observed 89 confusions, one insertion and no deletion. The main problem in the alphabet recognition task comes from confusions within a group of particularly error prone letters namely C, D, E, etc, which is similar to the English E-set.

During each iteration all reference utterances were used in random order. The constant slope of the sigmoid function was set to $\alpha = 10$ and for the I-th iteration an ϵ of

$$\epsilon = \epsilon_0 \cdot (1 - \frac{I}{MAX}) \qquad (13)$$

was used, MAX denoting the maximum number of iterations. In our simulation experiments we used the values of $\epsilon_0 = 0.001$ and $MAX = 200$.

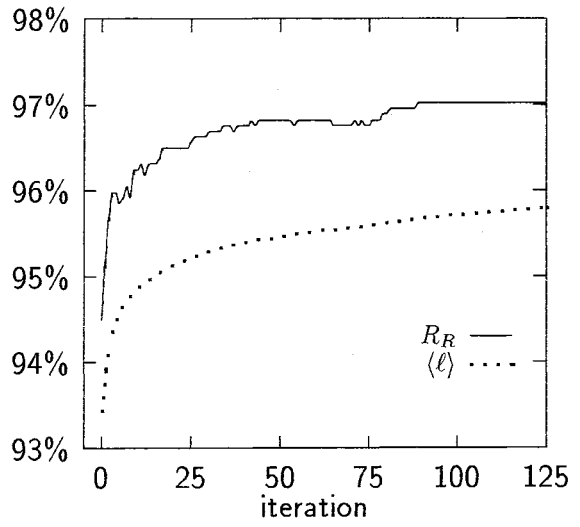

Figure 1: Recognition rate R_R and expected rate $\langle \ell(X) \rangle$ over the number of iterations

5 Simulation Results

In the first experiment we examined the potential of our approach in the case of identical dimension of the original and transformed feature vectors. Then the transformation matrix $\boldsymbol{H}$ resulting from the principal component analysis can be used directly as the start point of the GPD optimization. In order to isolate the effects of the optimized transformation matrix all other model parameters were held unchanged.

Figure 1 shows the resulting recognition rate R_R for the training data calculated after each iteration. Additionally, the average value $\langle \ell(X) \rangle$ over all utterances is included. As described above both R_R and $\langle \ell(X) \rangle$ are based on isolated word recognition of the segmented training utterances. Considering for example the initial word models this constrained recognition yields R_R=94.5% versus the original rate of R_R=93.3%. During the reestimation procedure the first iterations yield major improvements. For large I $\langle \ell(X) \rangle$ of the reference set still increases slowly. The recognition rate R_R, however, then shows only minor improvements.

Applying the optimized transformation matrix in the recognition of the test data resulted in a performance of R_T=90.4%, i.e. an improvement of 3.1 compared to the initial rate of R_T=87.3%. As well as in the initial system most of the errors are confusions within the E-set. Only 2 insertions and no deletion occurred. Again, the first few iterations yield most of the increase in R_T. E.g. the recognition rate after 15 iterations was already R_T=89.8%. Further improvements can be achieved by GPD-update of the HMM parameters. Reestimating the means μ_{il} and weights $b_c(l, q_n)$ resulted in R_T=92.3%.

In order to get more insights into the effects of the GPD reestimation we measured during the reestima-

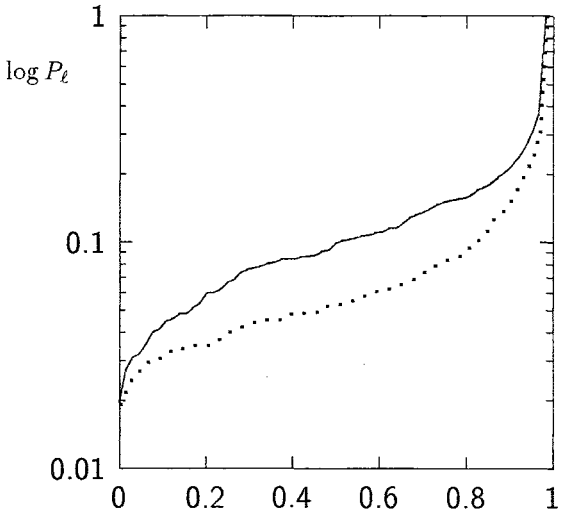

Figure 2: Distribution function $P_\ell(z)$ with initial models (—) and after 200 iterations (···)

tion process the distribution of the values of $\ell(X)$. As described above $\ell(X) \sim 1$ corresponds to a safe recognition and $\ell(X) \sim 0$ to a high error probability. In Figure 2 the distribution function $P_\ell(z)$ is given both for the start of the reestimation and after 200 iterations. During the reestimation utterances with $0.2 \leq \ell(X) \leq 0.8$ are most affected, yielding an increased number of recognitions with $\ell(X)$ close to one.

Next we extended the approach to the general case of different dimensions of the original and transformed feature vectors. As feature vector we used a concatenation of five consecutive input vectors. Each input vector contained 12 cepstral coefficients and one Δenergy coefficient. Using a matrix $\boldsymbol{T}$ that incorporates the calculation of the Δcepstral coefficients the resulting 65-dimensional vectors can be transformed back to the 25-dimensional vectors used so far. In this approach two effects can be distinguished. First, the GPD training can be applied to the weights therefore providing a seamless integration of these weights in a general training procedure. Second, reestimating all elements of the 65×25-matrix $\boldsymbol{A}$ leads to an implicit feature selection scheme.

Together with the principal component transformation $\boldsymbol{H}$ the transformation matrix

$$\boldsymbol{A} = \boldsymbol{H} \cdot \boldsymbol{T} \tag{14}$$

results. The matrix $\boldsymbol{T}$ has only few nonzero elements, namely the weights for the calculation of the Δcepstral coefficients and a subdiagonal of 1 in order to pass through the cepstral coefficients and the Δenergy coefficient. The structure of the matrix can easily be maintained in the GPD approach by only updating the nonzero elements. After the multiplication (14) all elements of $\boldsymbol{A}$, however, are in general nonzero and it is difficult to impose constraints on the reestimation.

Therefore we replaced in a first experiment the principal component transformation $\boldsymbol{H}$ by a diagonal matrix consisting of the inverse variances of the features measured over all training data. Using this matrix we trained new word models. The recognition performance was almost the same for both types of feature normalization. Optimizing the nonzero elements of $\boldsymbol{A}$ then yielded a recognition rate of R_T=90.2%.

Finally, we extended the GPD update to a full 65×25-matrix. In this case the number of parameters involved in the optimization increases significantly. Therefore the recognition rate on the training set improved much more and we found rates R_R of more than 99%. Nevertheless the performance on the test data of R_T=90.3% was almost the same as obtained with the 25×25-matrix.

6 Conclusion

In this paper we discussed the use of GPD-based minimum error training for linear feature transformation. This approach allows a seamless integration of the transformation matrix into a general training procedure. The results show that applying the proposed training procedure results in an improved recognition performance. The recognition of letters in spelled names improved from 87.3% to 90.4%. In our case the use of large input vectors build from consecutive cepstral vectors did not lead to further improvements. Further experiments are necessary to examine whether this effect is due to an insufficient amount of training data.

The minimum error training approach provides a theoretical framework that allows inclusion of at least part of the feature extraction in theoptimization process. The described integrated training of the linear transformation matrix is one further step towards a unified training scheme that covers all steps from preprocessing to language modeling.

References

[1] F. Class, A. Kaltenmeier, and P. Regel-Brietzmann. Optimization of an HMM–based continuous speech recognizer. In *EUROSPEECH 93*, pages 803–806, Berlin, 1993.

[2] E. G. Schukat-Talamazzini, J. Hornegger, and H. Niemann. Optimal linear feature transformations for semi–continuous hidden Markov models. In *ICASSP-95*, Detroit, 1995.

[3] L. Deng. Integrated optimization of dynamic feature parameters for hidden Marov modeling of speech. *IEEE Signal Processing Letters*, SPL-1:66–69, 1994.

[4] W. Chou, B.H. Juang, and C.H. Lee. Segmental GPD training of HMM based speech recognizer. In *ICASSP–92*, pages I–473–476, San Francisco, 1992.

[5] S. Euler and J. Zinke. Experiments on the use of the generalized probabilistic descent method in speech recognition. In *ICSLP-92*, pages 157–160, Banff, 1992.