Improving Speaker Recognition Performance Using Phonetically Structured Gaussian Mixture Models

R. Faltlhauser, G. Ruske

Inst. for Human-Machine-Communication
Technische Universität München (TUM), Munich, Germany
{Faltlhauser,Ruske}@ei.tum.de

Abstract

Throughout the past few years it has been shown that Gaussian Mixture Models (GMM) are highly suitable for speaker identification and verification. Nevertheless these models try to represent primarily the distribution of the available training data - neglecting any possible phonetic information which might be of worth. In our paper we present a recognition system using multiple speaker GMMs based on phonetic classes. By introducing 'phonetic' mixture coefficients a weighting of phoneme classes with respect to speaker recognizability can be achieved. The implicit integration in the probability computation avoids the need for a phonetic labeling during recognition. The mixture weights can be learned in a training phase. Model training was examined applying MAP enrolment as well as the recently reported Eigenvoice approach. Especially for the latter the phonetic separation has shown to be advantageous. Recognition error reductions up to 15 % relatively were achieved. Furthermore, the multiple GMM approach is particularly effective for speaker enrolment with sparse training data.

1. Introduction

Throughout the past years Gaussian Mixture Models (GMM) [1] have proven to be highly capable for identifying or verifying speakers - commonly superior to Hidden Markov Models (HMM) or other classifiers [1, 2]. Although they do not directly consider phonetic information, the densities contained in a GMM do reflect the whole phonetic space - if training data is abundant. Commonly speaker dependent models are generated by retraining a speaker independent model, which should cover the whole phonetic space. However, if only few training data are available, some phonemes are sparsely represented or even will be missing totally in the enrolment utterances. In this case the densities covering those phonemes will probably be negatively affected by the data of some other phonemes - leading to a potentially worsened modeling for this speaker.

In our approach we tried to constrain the influence of phonemes on each other by modeling each phoneme or phonetic class by an individual GMM. Since the timing information is of minor importance for the task of speaker identification, every speaker can be represented by multiple GMMs (instead of HMMs) - one per phonetic class. Another way of imposing constraints on the degree of freedom was presented by Thyes [3] et al. They confine model parameters by using the Eigenvoice method, which is basically a subspace projection technique. In our system we tried to combine both methods.

Moreover, in past studies it has been shown, that speaker discrimination varies among the different phonemes or phoneme categories [2, 4, 5]; e.g. Parris [5] used HMMs to determine the usefulness of the individual phonemes. Table 1 depicts speaker identification performance for some sample phonemes and their frame frequency in the training data normalized on the most frequent phoneme.

Phoneme	#occur. (rel.)	recognition rate (percent)
/n/	1.00	32.7
/m/	0.53	29.2
/s/	0.85	20.8
/p/	0.09	8.3
/f/	0.37	11.1
/a/	0.30	19.1
/e:/	0.20	17.9
/x/	0.16	10.6

Table 1: Speaker recognition performance on frame level for some sample phonemes (74 speakers, German Verbmobil database).

In accordance with previous studies Table 1 clearly shows that even some of the minor occurring phonemes possess a high speaker separating power. In order to account for the varying speaker separating properties of phonemes Auckenthaler [2] used a phonetic weighting of the GMM scores. In our approach we tried to exploit this source of information by introducing mixture weights for the different phonetic classes. These weights can be learned automatically by optimizing a confidence measure during the training phase. Furthermore there is no need for a phonetic labeling during recognition.

2. Phonetically Structured GMMs

2.1. Parallel Gaussian Mixture Models

In a standard GMM based speaker identification system setup each speaker is modeled by a single, individual GMM. In our approach we represent each speaker by several GMMs - each of them covers an individual phoneme or a group (class) of phonemes. Fig 1 illustrates the system setup. The separation of phonemes is especially useful during speaker enrolment. For the recognition phase the models can also be recombined - as explained later on.

Speaker enrolment is commonly done by retraining a general speaker independent background model using training algorithms like ML or MAP [8]. If enrolment data are scarce,

preferably the MAP algorithm is applied in order not to affect models of unseen data, for which the parameters of a background model are usually better trained. Nevertheless, if some phonemes are only sparsely contained in the training data, the Gaussian densities of the background model representing these phonemes are affected by other phonemes contained in the data. This would lead to a detriment in modeling, especially if - for some other speakers - these phonemes are seen in the training data. A separation of the individual phonemes or groups of phonemes could at least partially prevent such negative influences.

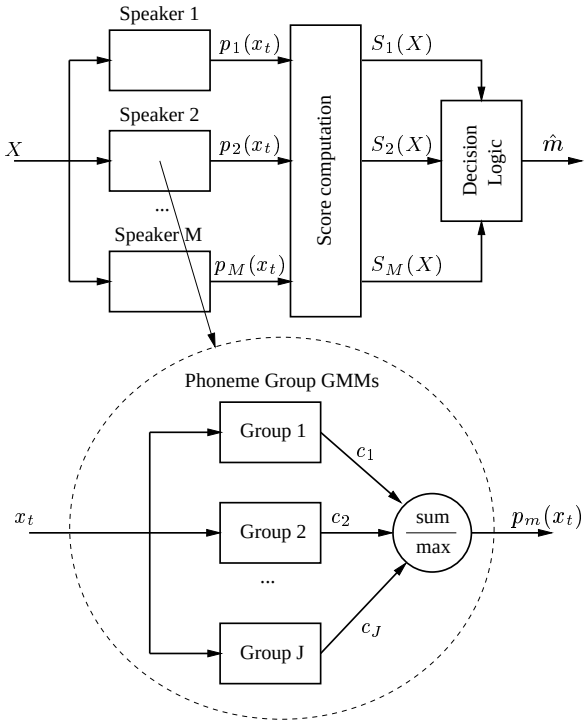

Figure 1: Parallel phoneme group GMMs for a speaker.

Each phonetic class model M_{mj} of speaker m is represented by a Gaussian mixture distribution with K_j densities:

$$p(x|m,j) = \sum_{k=1}^{K_j} c_{mjk} \mathcal{N}(x, \mu_{mjk}, \Sigma_{mjk},)$$

The mixture coefficients for phonetic class model j of a speaker m have to obey the probabalistic constraint:

$$\sum_{k=1}^{K_j} c_{mjk} = 1$$

The probability for speaker m is obtained by summing over all submodels j.

$$p_m(x) = p(x|m) = \sum_{j=1}^{J} p(x|m,j)$$

In order to take into consideration that different phonemes have different speaker discriminating properties [4] we introduced a speaker independent weighting factor c_j for each class (see Fig. 1) which can be interpreted as a 'phonetic mixture coefficient'. Therefore, $p(x|m)$ becomes

$$p(x|m) = \sum_{j=1}^{J} c_j p(x|m,j)$$

Like the mixture coefficients in each class model, these further mixture coefficients have to obey the probabalistic constraint:

$$\sum_{j=1}^{J} c_j = 1$$

An approximation for the speaker probability is obtained by taking the maximum probability of all submodels j (s. Fig 1):

$$p(x|m) = \max_j c_j p(x|m,j)$$

Using the maximum approximation, the speaker GMMs are treated as parallel GMMs. In case of summing over all individual model outputs $p(x|m,j)$ the individual GMMs can - for the recognition phase - basically be recombined giving a single GMM with $K^* = \sum_{j=1}^{J} K_j$ densities and modified mixture coefficients:

$$c^*_{k*} = c_j c_{mjk}$$

Nevertheless, the main advantage of the approach lies in the separate training of the Gaussian distributions for groups of phonemes on the one hand and the incorporation of phonetic mixture coefficients on the other hand. As explained in section 3 these coefficients can be independently learned in a training phase. Because of the implicit nature of the mixture weights there is no need for a phonetic labeling during recognition.

2.2. Phoneme Classes

The selection of phoneme classes is based on several factors:

- phonetic similarity
- speaker separability of the individual phonemes
- frequency of occurrence

As shown later in the results section we experimented with various class sizes ranging from 10 to 16 classes. Broad classes like nasals, fricatives, different vowels, liquids as well as plosives are forming base classes which were subdivided giving 10 to 16 groups overall. E.g. the nasal /m/ was selected as a single class, since it offers a high frequency (s. Table 1) together with good speaker separability. Other phonemes showing only few occurrences were combined with phonetically similar phonemes. Table 2 displays some sample classes.

class	Phonemes
nasals 1	/m/
nasals 2	/n/ /N/
fricatives	/s/ /S/ /z/ /Z/ /f/ /v/ /C/ /x/ /j/
liquids	/l/ /r/
plosives	/t/ /k/ /d/ /b/ /p/ /g/
vowels 1	/E:/ /e:/ /E/

Table 2: Class composition for some sample classes.

2.3. Model Training

In order to train the phonetic GMMs two major issues have to be examined:

- Segmented data
- Model size (number of Gaussians) per submodel

A key issue for the training of proper phonetic models is the availability of "pure" data for each phoneme. For this purpose we were using fully trained HMM models of an automatic speech recognition system for the phonetic segmentation of the training utterances. Moreover we applied pronunciation variant modeling in order to find a more exact mapping between phonemes and transliteration. The transliteration of the utterance was assumed to be known. Since the phonemes are occurring with different frequencies each class has to be modeled with a proper number of Gaussian densities. For this purpose we applied a likelihood based cluster algorithm, similar to the one described in [6], which was already successfully utilized for HMM codebook initialization. The algorithm is basically an extension to the well-known LBG-algorithm: through iterative splitting of Gaussian mixtures the number of prototypes is continuously increased until some likelihood based termination criterion is satisfied. This way the number of Gaussian densities per submodel can be determined automatically.

Speaker dependent models were created from a speaker independent background model. For the multiple GMM setup individual background models were trained for all phonetic classes. Model enrolment was performed using either the Maximum Aposteriori (MAP) [8] training algorithm or the recently reported Eigenvoice [3, 7] approach. The Eigenvoice method is basically a projection onto a precomputed subspace, which constrains the degree of freedom. The method should be particularly effective if only few enrolment data are available.

3. Class Mixture Coefficients

A major advantage in training parallel GMMs is the fact, that the class mixture coefficients c_j can be determined separately. We investigated several approaches of selecting/computing the class weights $\mathbf{c}$:

- Equal coefficients
- Class A-priori probability
- Optimization of confidence measure $L(X, \mathbf{c})$

For the latter case we determined $\mathbf{c}$ by optimizing the confidence measure:

$$L(X, \mathbf{c}) = \log p(X|\bar{m}) - \log p(X|\hat{m}) =$$

$$= \sum_T \log p(x_t|\bar{m}_t) - \sum_T \log p(x_t|\hat{m}_t)$$

$$L(X, \mathbf{c}) = \sum_T \log \sum_{j=1}^{J} c_j p(x_t|\bar{m}_t, j)$$

$$- \sum_T \log \sum_{j=1}^{J} c_j p(x_t|\hat{m}_t, j)$$

This measure of confidence evaluates the score distance between the correct speaker $\bar{m}$ and a rival model $\hat{m}$. Since the GMM parameters for each class GMM, i.e. the probabilities $p(x_t|m, j)$ are trained in advance, the above confidence criterion is only dependent on $\mathbf{c}$. As rival models (on frame basis) we examined 2 possibilities:

- Top-1 rival speaker
- Background model

For the optimization of $L(X, \mathbf{c})$ gradient descent techniques can be applied.

4. Experimental Results

In order to test our approach we used two experimental setups. The first setup comprised 74 male and female speakers with sufficient enrolment data available (>30 seconds per speaker). Utterances for training were taken from the German Verbmobil Spontaneous Speech Corpus. Preprocessing uses 12 Mel-cepstral coefficients. All speaker GMMs have 64 Gaussian densities with diagonal covariance matrices. In the multi GMM approach the total number of Gaussians per speaker is also limited to 64. Speaker dependent models were created using MAP for the retraining of a speaker independent background model. The baseline system had 1 GMM per speaker, whereas the multiple GMM setup comprised 10 GMMs. Phonetic mixture coefficients for each phoneme group were created as described in section 3. Table 3 shows the results obtained. In the multiple GMM section of Table 3 the leftmost column depicts whether parallel ('max') or a recombined single GMM ('sum') was used. The second column indicates the training mode for the phonetic mixtures.

Adaptation method		recognition rate (in percent) 1 sec.	3 sec.
single GMM (baseline)			
MAP		87.47	96.21
MAP, Means only		88.12	96.21
mult. GMMs, 10 Groups			
max	equal	88.77	97.19
	apriori	89,78	97,19
	top-1	89.33	96,63
	background	89.54	96.91
sum	equal	88,69	97,47
	apriori	89.45	97.19
	top-1	89.41	97.19
	background	89.45	97.19

Table 3: Recognition performance for 1 and 3 seconds test condition, enrolment data > 30 seconds.

Table 3 shows an error reduction up to 15.8% relatively to the baseline system. Although the use of multiple GMMs per speaker already gave a reduction of 10.4% relatively, the application of adapted mixture weights showed a further error rate reduction. In Table 4 the mixture coefficients for the sample classes (s. Table 2) are displayed. Astonishingly, the nasal /m/ ('nasals 1') was weighted down by the algorithm, although it has a high capability for separating speakers (see Table 1).

The multiple GMM approach should be particularly effective if training data are scarce. In order to examine this task we enroled 26 speakers not contained in the first experiment. Training data was limited to a 5 second enrolment per speaker. Background models as well as phonetic mixture weights were taken from the first experiment.

Phoneme	top-1	background
nasals 1	0.049	0.072
nasals 2	0.127	0.152
fricatives	0.243	0.160
liquids	0.092	0.078
plosives	0.268	0.147
vowels 1 (e)	0.034	0.074

Table 4: Mixture weights for sample classes.

Adaptation method		recognition rate (in percent) 1 sec.	3 sec.
single GMM (baseline)			
MAP		70.40	86.07
MAP, Means only		68.43	84.14
mult. GMMs, 10 Groups			
max	equal	72.61	85.66
	apriori	74,70	86,90
	top-1	73.37	87.03
	background	73.82	86.62
sum	equal	73.52	85.93
	apriori	74.62	87.45
	top-1	74.58	87.31
	background	74.77	87.59
mult. GMMs, 16 Groups			
max	equal	72.00	84.97

Table 5: Recognition performance for 1 and 3 seconds test condition, 26 speakers, 5 seconds speaker enrolment.

Table 5 shows error rate reductions up to 15.8% relatively, depending on the evaluation time and weighting mode. Whereas for the 1 second test an improvement was achieved regardless of the weighting applied, for the 3 second test only the introduction of an adequate weighting led to error rate reductions up to 12.8% relatively. The use of 16 phonetic classes brought a slightly worse recognition performance. As a further comparison we used the Eigenvoice approach for group enrolment. A thorough description of this algorithm can be found in [7]. We computed the Eigenspace (PCA) for each group individually using the GMM models from the first experiment. Only the mean parameters were included. Speaker enrolment was performed by means of Maximum Likelihood Eigen-Decomposition (MLED). In case of multiple GMMs, the dimension of the individual subspace (#EV), is limited by the dimension of the original space: $K_j * Dim_{MFCC}$. The Eigenvoice approach should be compared with the 'MAP, means only' baseline system, since only the Gaussian mean parameters are altered. Table 6 shows for the multi-GMM approach improved recognition rates in comparison to the single GMM setup. This effect could be related to the fact, that in the multiple GMM case an individual subspace weighting vector is used for each phonetic submodel, whereas in the single GMM case a single mutual weighting vector is used for the whole GMM, i.e. for all 'phonetic' densities. This seems too rigid for the modeling of speaker specific characteristics.

5. Summary

In our paper we propose a GMM training approach based on phonetic classes in combination with a phonetic weighting. Depending on the task, recognition error reductions up to 15% relatively are achieved. For each class an individual GMM is trained. For the recognition phase these models can also be recombined into a single GMM. By the separation of phonetic classes there is a fix relationship between a Gaussian density and the phoneme or phonetic class, which it is representing. By this way densities representing unseen phonemes are not detrimented by other classes appearing in the training data. So the robust estimation of the initial speaker independent model can be maintained. This approach seems particularly useful for fast speaker enrolment. Moreover, for the Eigenvoice adaptation method the splitting into phoneme classes brought particular improvement, since the probably too rigid subspace constraints for entire single GMMs are lessened.

EigenVoices (EV)		recognition rate (in percent) 1 sec.	3 sec.
#EV		single GMM (baseline)	
20		61.36	76.00
30		65,46	79,31
40		67.14	80.14
50		67.06	79.17
70		65.46	78.34
weights	#EV	mult. GMMs, 10 Groups	
max, equal	2	38.18	54.21
	10	68.28	83.86
	20	70.55	84.14
	30	71.69	84.69
	40	70.06	84.28
	50	70.29	83.45
max, apriori	30	73.18	86.34

Table 6: Recognition performance for model training using Eigenvoices (26 speakers, 5 seconds enrolment).

6. References

[1] D. A. Reynolds, R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", IEEE Trans. Speech and Audio Processing, vol. 3, pp. 72-83, 1995.

[2] R. Auckenthaler, E.S. Parris, M.J. Carey, "Improving a GMM Speaker Verification System by Phonetic Weighting", Proc. ICASSP 99, paper no. 1440.

[3] O. Thyes, R. Kuhn, P. Nguyen, J.C. Junqua, "Speaker Identification and Verification using Eigenvoices", Proc. ICSLP 2000, paper no. 1155.

[4] J. Eatock, J. Mason, "A Quantitative Assessment of the Relative Speaker Discriminative Properties of Phonemes", Proc. ICASSP 94, pp. 1133-1136.

[5] E. S. Parris, M. J. Carey, "Discriminative Phonemes for Speaker Identification", Proc. ICSLP 94, pp. 1843-1846.

[6] R. Faltlhauser, T. Pfau, G. Ruske, "Creating Hidden Markov Models for Fast Speech by Optimized Clustering", Proc. Eurospeech 99, pp. 1355-1358.

[7] R. Kuhn, J. C. Junqua, P. Nguyen, N. Niedzielski, "Rapid Speaker Adaptation in Eigenvoice Space", IEEE Trans. Speech and Audio Processing, vol. 8, pp. 695-707, 2000.

[8] J.-L. Gauvain, C.-H. Lee, "Improved Acoustic Modeling with Bayesian Learning", Proc. ICASSP 92, pp. 481-484.