Combination of vector quantization and Gaussian mixture models for speaker verification with sparse training data

Guido Kolano *Dr. Peter Regel-Brietzmann*

DaimlerChrysler AG, Research and Technology, Wilhelm-Runge-Str. 11,
D-89081 Ulm, Germany
e-mail: {guido.kolano, peter.regel-brietzmann}@daimlerchrysler.com

10.21437/Eurospeech.1999-308

ABSTRACT

We present a combination of an extended vector quantization (VQ) algorithm for training a speaker model and a gaussian interpretation of the VQ speaker model in the verification phase. This leads to a large decrease of the error rates compared to normal vector quantization and only a slight deterioration compared to full Gaussian mixture model (GMM) training. The training costs of the new method are only slightly higher than for pure vector quantization.

1. INTRODUCTION

In automatic speaker verification we have to find a tradeoff between user comfort (i.e. small amount of training data, fast training) and low error rates (this normally implies large amount of training data from every speaker in the system). GMMs, as introduced by Reynolds [4], perform very well but training requires a lot of time and they get numerically unstable when trained with small amount of data. The main problem is the inversion of the (underestimated) covariance matrices. Pure vector quantization, e.g. using k-means clustering or the LBG algorithm by Linde, Buzo and Gray [3], on the other hand is numerically stable and rather fast, but the performance in speaker verification is not as good as for GMMs.

2. BASELINE SYSTEMS

The new method, which is proposed in the next section, will be a combination of vector quantization (VQ) and Gaussian mixture modelling (GMM). Therefore we briefly want to introduce our versions of these methods.

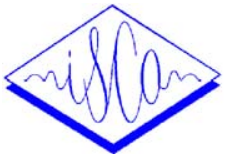
For VQ training we use a deterministic version of the LBG-algorithm with an Euclidean distance measure. In opposite to the original LBG-algorithm [3] the splitting of the clusters is not done by random variation of the mean vectors. We modify only the component of the meanvector with the largest variance. The variations of this component are $\pm 20\%$ of the standard deviation of the component.

The utterance score $D_{utt}(\vec{X}|\lambda)$ is calculated as the arithmetic mean of the N Euclidean distances $d_{frame}(\vec{x_i}|\lambda)$ between the feature vectors $\vec{x_i}$ and the nearest center of the evaluated speaker (or background) model λ to frame i:

$$D_{utt}(\vec{X}|\lambda) = \frac{1}{N}\sum_{i=1}^{N} d_{frame}(\vec{x_i}|\lambda)$$

Our GMM-system uses the standard EM algorithm as described by Reynolds [4]. To improve the stability of the algorithm, we set a lower limit for the absolute of every component of the covariance matrices. We use nodal complete covariance matrices (one complete covariance matrix per mean vector). We also tried diagonal covariance matrices, but the error rates with diagonal matrices are allways higher even if we take more distributions (mean vectors). A global covariance matrix also doesn't yield good results. The initialisation of the mean vectors is done by our VQ algorithm. This seems to be more time consuming than random initialization, without getting recognition error rates [4]. But we have two arguments for doing this:

1. In our experiments we also estimate the VQ parameters for comparison, so no extra calculation has to be done by this initialization.



6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

2. We are not in danger to obtain an initialization vector, which lies far away from all other vectors and has a badly estimated covariance matrix, which can't be inverted. So we can use higher orders for the GMM without getting problems when inverting the covariance matrices. With the VQ initialization we get similar weights for all distributions.

For one frame with a D-dimensional feature vector x_i the likelihood of an GMM with order M is:

$$p_{frame}(x_i|\lambda) = \sum_{j=1}^{M} w_j b_j$$

with

$$b_j = \frac{1}{(2\pi)^{D/2}|\sigma_j|^{1/2}} e^{\left(-\frac{1}{2}(\vec{x_i}-\vec{\mu_j})^T \sigma_j^{-1}(\vec{x_i}-\vec{\mu_j})\right)}$$

with the weights w_j, the mean vectors μ_j and the covariance matrix σ_j.

The utterance score $P_{utt}(\vec{X}|\lambda)$ in the GMM case for a given model λ is the loglikelihood for all N frames x_i as described in [5]:

$$P_{utt}(\vec{X}|\lambda) = \frac{1}{N}\sum_{i=1}^{N} p_{frame}(\vec{x_i}|\lambda)$$

For VQ as well as GMM we train a background model in order to normalize the scores of the speaker models. This model has the same structure as the speaker models but it is trained with data from 33 background speakers. Background normalization is done in both cases by calculating the difference between the utterance score calculated by the speaker model and the utterance score calculated by the background model (for GMM this is known as likelihood normalization). If the difference is less than a given threshold, the speaker is rejected, otherwise the speaker is accepted.

Doing background normalization in VQ, some people use a different normalization and calculation of utterance scores: they use the count how often the speaker model fits better than the background model and take the relative frequency as score. In our earlier experiments this, however, led to higher error rates.

3. PROPOSED METHOD

The idea for the combination of VQ and GMM is based on the fact, that both methods try to represent the distribution of data vectors in the feature space. The way of representing the distribution is different, but there are obvious similarities: for both methods we get centers/mean vectors and for both methods we can give a measure for the weight of being associated (hard decision at VQ, soft decision at GMM).

Our proposed method uses a combination of the described VQ method for training and the GMM method for evaluation. The problem is, that the GMM models have more parameters than the VQ models with the same order (number of centers). Both methods need the mean vectors μ_j (centers), but for GMM we additionally need the corresponding covariance matrices σ_j and weights w_j. In order to estimate these parameters we do one additional step after calculating the VQ parameters: for every feature vector in the training set we look for the center with the least distance and associate it with the vector. Then we calculate for every center the covariance matrix of the associated vectors and the relative frequency of being associated. The relative frequency is an estimation of the weight of a center and the covariance matrices are taken as estimation for the GMM covariance matrices. Having done this, we have estimations of all necessary parameters of the GMM. In the verification phase we treat our new model like a regularly trained GMM. In this paper we refer to this method as EVQ (extended vector quantization).

In the framework of GMM you can see this method as an initialization of a GMM without the EM training. It is obvious, that this simpler model cannot describe the distribution of the vectors as good as a fully trained GMM. Our experiments show, however, that you get much better recognition rates than simply performing VQ, but we have only a small degradation compared to full GMM training.

Compared to GMM training the EVQ training has two differences:

1. The distance measure for EVQ is Euclidean,

for GMM a Mahalanobian distance measure is used.

2. The estimation of mean vectors and covariance matrices in the EVQ method is based on the data vectors associated to a center only. Moreover all associated data vectors are weighted equally. This leads to a segmentation of the feature space. In the GMM case all data vectors are used for the estimation of every mean vector and covariance matrix, but the data vectors are weighted. No segmentation of the feature space is done.

4. SPEECH DATA AND PREPROCESSING

In our experiments we used the 106 male speakers from the YOHO speaker verification database [2]. This database is divided into 2 sections for enrollment (training) and verification (testing). The enrollment data have been recorded in 4 sessions with 24 "combination-lock" phrases (like "24-72-54") each, the verification data in 10 sessions with 4 utterances each. So we have 96 utterances for training and 40 utterances for testing. The sampling frequency is 8 kHz, the resolution is 12 bits. All recordings have been done in a real-world office environment.

Preprocessing is done by the *cepstrum* programme of He's *spkrtool* [1]. He's *spkrtool* is a collection of programms for speaker identification with different methods. It is freely available on the Web. In our experiments we use 16 MFCC + pitch, calculated from the voiced parts of an utterance. So we get 17 coefficients per frame. The frames start at every 8 ms and are 16 ms long.

5. EXPERIMENTS AND RESULTS

First we want to give some more detailed information about our training methods. For VQ we carried out for every step so much iterations until no more changes in the mean vectors appeared, i.e. until no data vector has changed its associated center. Then we doubled the number of centers as described. In the GMM training we stopped the EM algorithm after 10 iterations.

Using higher orders, similar problems occur for the EVQ method as they are for the GMM method: the covariance matrices are underestimated and not necessarily invertable. In these cases we use two methods: first we set a small limit for the components of the covariance matix and if that doesn't help, we exclude those centers, whose corresponding covariance matrices can't be inverted. This leads to a somewhat smaller effective order, but the excluded centers have only very few data vectors associated and the error we provoke is neglectable. This exclusion is done after training the EVQ model, and unlike in GMM training we don't need the covariance matrices during training. With our data and preprocessing, we are able to train GMMs up to an order of 16 and EVQ up to an order of 64 without major problems due to underestimation of covariance matrices. Standard VQ could be done up to even more higher orders, but in those cases no covariance matrices can be estimated because we don't have enough data vectors associated to the centers.

To get independent background speakers, the speakers with YOHO-IDs less than 200 (73 speakers) are taken for training and testing the models, the data from the remaining 33 speakers are used to train the background model. These 33 speakers are not used for testing.

For comparable results we use the Equal Error Rate (EER). This means that both the false rejection rate (FR) and the false acceptance rate (FA) are set equal by setting the threshold after getting all utterance scores.

We use two different ways of thresholding:

1. Every speaker has his own threshold, i.e. for every speaker the threshold is set in such a way that the error rates are equal. We call this *local thresholding*.

2. All speakers have the same threshold, i.e. for a single speaker the error rates FA and FR are not equal, but if you look at all speakers the mean of FA and FR is equal. We call this *global thresholding*.

Due to the nonlinear dependence of the error

method	model order (No. of centers)				
	4	8	16	32	64
VQ	11.4	9.6	9.0	6.8	5.4
GMM	4.3	3.1	2.3	—	—
EVQ	5.4	4.2	3.3	2.8	2.8

Table 1: Equal error rates (EER) in percent for speaker verification (local thresholding)

method	model order (No. of centers)				
	4	8	16	32	64
VQ	12.9	11.1	9.6	8.1	6.9
GMM	5.3	4.1	3.3	—	—
EVQ	6.3	5.2	4.5	5.9	4.2

Table 2: Equal error rates (EER) in percent for speaker verification (global thresholding)

rates FA and FR on the threshold, local thresholding leads to lower mean error rates. On the other hand we have to estimate the threshold for every new speaker. When adding a new speaker to the system, local thresholding causes data from other speakers (imposters) to estimate the threshold. The price for better verification results is more memory to store the speech data and the time for estimating the threshold.

The results of our experimets are given in table 1 and table 2. Using local and global thresholding the situation is similar: taking the EVQ model as a GMM, we can reduce the error rates dramatically. On the other hand the EVQ error rates are about 1% higher than the GMM error rates at the same order.

6. CONCLUSION

A new combination of vector quantization and gaussian mixture models has been proposed introduced, which reduces the error rates of VQ models to nearly GMM level without the additional costs of GMM training and the numerical unstabilities when using sparse training data.

We show that the similarities between VQ and GMM can be used to combine the advantages of both methods: the fast training procedure of the VQ algorithm and the good verification performance of GMM. The costs of the faster training is a slight increase in the error rates compared to GMM, but the results are much better than pure VQ.

REFERENCES

1. Jialong He's *spkrtool* and a short description are available at *http://www.speech.cs.cmu.edu/comp.speech/Section6/Verification/jialong.html*

2. A.Higgins, J.Porter, K.Bahler: *YOHO Speaker Authentication Final Report*, ITT Defense Communications Division, 1989

3. Y.Linde, A.Buzo, R.M.Gray: *An algorithm for vector quantizer design*, IEEE Trans. on Communications **28**(1980)84-95

4. D.A.Reynolds, R.C.Rose: *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*, IEEE Trans. on Speech and Audio Processing **3**(1995)72-83

5. D.A.Reynolds: *Speaker identification and verification using Gaussian mixture speaker models*. Speech Communication **17**(1995)91-108