Pattern Recognition Letters 30 (2009) 341–347

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Comparative evaluation of maximum *a Posteriori* vector quantization and gaussian mixture models in speaker verification

Tomi Kinnunen, Juhani Saastamoinen, Ville Hautamäki *, Mikko Vinni, Pasi Fränti

Speech and Image Processing Unit (SIPU), Department of Computer Science and Statistics, University of Joensuu, P.O. Box 111, FI-80101 Joensuu, Finland

ARTICLE INFO

Article history:
Received 19 July 2008
Available online 27 November 2008

Communicated by R.C. Guido

Keywords:
Speaker verification
MFCC features
Gaussian mixture model (GMM)
Vector quantization (VQ)
Universal background model (UBM)
Support vector machine (SVM)

ABSTRACT

Gaussian mixture model with universal background model (GMM–UBM) is a standard reference classifier in speaker verification. We have recently proposed a simplified model using vector quantization (VQ–UBM). In this study, we extensively compare these two classifiers on NIST 2005, 2006 and 2008 SRE corpora, while having a standard discriminative classifier (GLDS–SVM) as a point of reference. We focus on parameter setting for *N*-top scoring, model order, and performance for different amounts of training data. The most interesting result, against a general belief, is that GMM–UBM yields better results for short segments whereas VQ–UBM is good for long utterances. The results also suggest that maximum likelihood training of the UBM is sub-optimal, and hence, alternative ways to train the UBM should be considered.

1. Introduction

Typical speaker verification systems use mel-frequency cepstral coefficients (MFCCs) to parameterize speech signal. The features are usually processed with some form of feature normalization or transformation to enhance robustness against channel variability. A voice activity detector (VAD) is also needed for rejecting non-speech frames.

A speaker model is created from the extracted features. In the 21st century, two approaches have been dominant for speaker modeling. The first one, based on generative modeling, uses *maximum a posteriori* (MAP) adaptation of a speaker-independent *universal background model* (UBM) (Reynolds et al., 2000; Hautamäki et al., 2008). The second approach, based on discriminative training, finds the parameters of a hyperplane that separates the target speaker from a set of background speakers (Campbell et al., 2006a). A recent trend are *hybrid* models that combine the good properties of both approaches (Campbell et al., 2006b; Lee et al., 2008). For instance, in the GMM supervector method (Campbell et al., 2006b), the MAP-adapted mean vectors are stacked to form a single, high-dimensional feature vector for the utterance. These supervectors are then treated as feature vectors when training an SVM. Latest solutions also use a so-called *eigenchannel* transformation and *joint factor analysis* (JFA) to reduce the effects of channel and session variability in the speaker models (Burget et al., 2007; Kenny et al., 2008; Vogt and Sridharan, 2008).

We use MFCCs and focus on the speaker modeling component (classifier). We consider the following speaker modeling techniques:

(i) Gaussian mixture model with UBM (GMM–UBM) (Reynolds et al., 2000),
(ii) Vector quantizer with UBM (VQ–UBM) (Hautamäki et al., 2008),
(iii) Generalized linear discriminant sequence support vector machine (GLDS–SVM) (Campbell et al., 2006a).

We set the following limitations in order to keep the baseline simple: (1) we use only telephone data for background modeling, (2) we do not use any intersession variability compensation, (3) we do not make use of ASR component, (4) we do not make use of language information, (5) we do not use additional score normalization on top of UBM normalization, such as T-norm. More complete systems used in recent NIST speaker recognition evaluations (SRE) use such techniques in conjunction with each other. Our simplifications allow us to focus more deeply on the modeling component, but on the other hand, weaken the overall performance in comparison to more complete systems, especially for non-telephony data.

* Corresponding author. Fax: +358 132517955.
E-mail addresses: tkinnu@cs.joensuu.fi (T. Kinnunen), juhani@cs.joensuu.fi (J. Saastamoinen), villeh@cs.joensuu.fi (V. Hautamäki), mvinni@cs.joensuu.fi (M. Vinni), franti@cs.joensuu.fi (P. Fränti).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.11.007

Vector quantization speaker modeling was popular in the 1980s and 1990s (He et al., 1999; Soong et al., 1987), but after the introduction of the background model concept for GMMs (Reynolds et al., 2000), GMM has been the dominant approach. Even so, usually only the mean vectors of the GMM are adapted while using shared (co)variances and weights for all speakers. This raises a question whether the variances and weights are needed at all. To answer this question, we derived MAP adaptation algorithm for the VQ model (Hautamäki et al., 2008) as a special case of the MAP adaptation for GMM, involving only the centroid vectors. Posterior probabilities needed in the adaptation are estimated based on hard quantization with nearest neighbor classification. The VQ approach achieves speed-up in training compared to GMM with comparable accuracy.

In this paper, we further explore the inherent differences of the GMM–UBM and the VQ–UBM classifiers in the speaker verification task, while having the GLDS–SVM classifier as a point of reference. The results presented here are based on our submissions to NIST 2006 and NIST 2008 speaker recognition evaluations. We focus on parameter setting for fast N-top scoring, model order, performance for different amounts of training data and effects of mismatched data. In (Hautamäki et al., 2008), our main focus was in formal derivation of the algorithm rather than in extensive testing. This paper serves for that latter purpose.

Since the VQ model has less free parameters to be estimated, it may be hypothesized that VQ-based classifier will outperform GMM for small amounts of data; see, for instance, (David, 2002) for such an observation. This hypothesis is probably true if both models are trained using maximum likelihood (mean square error minimization). However, it is less clear how the situation changes when using MAP training for both models. In this paper, we will show surprising experimental evidence that suggests the opposite: GMM–UBM is better for short utterances whereas VQ–UBM outperforms GMM–UBM when the length of training and test data increases. We discuss the possible reasons for this and its implications.

In the following, we first describe in Section 2 the system components and the development datasets for optimizing the systems. The performance of the classifiers is then studied in Section 3, with the motivation of optimizing their parameters. Namely the number of components used in VQ and GMM, and the N-top parameter in the UBM. The best parameters are then applied for the NIST 2008 data in Section 4, followed by more detailed discussion of the GMM–UBM and VQ–UBM classifiers in Section 5. Finally, conclusions are drawn in Section 6.

2. System description

2.1. Feature extraction

Our front-end processing is similar to other systems used in NIST evaluations, such as (Reynolds et al., 2005) and (Tong et al., 2006). The MFCCs are extracted from 30 ms Hamming-windowed frames with 50% overlapping. First, 12 MFCCs are computed using a 27-channel mel-filterbank. The MFCC trajectories are then smoothed with RASTA filtering (Hermansky and Morgan, 1994), followed by computation and appending of the delta and double delta features. A FIR kernel $h = [-1, 0, 1]$ is used for obtaining the deltas. Double-deltas are obtained by applying the same kernel to the delta features. The last two steps are voice activity detection (VAD) and utterance-level mean and variance normalization in that order.

We use an adaptive, energy-based algorithm for VAD that uses a file-dependent detection threshold based on maximum energy level of the file. For completeness, we provide the Matlab code fragment in the following:

```
E = 20*log10(std(Frames')+eps); % Energies
maxl = max(E); % Maximum
I = (E>maxl-30) & (E>-55); % Indicator
```

2.2. Classifiers

GMM–UBM system follows the standard implementation (Reynolds et al., 2000). Diagonal covariance matrices are used in the mixture components. Two gender-dependent UBMs are trained. A deterministic splitting method is used for initializing the mean vectors, followed by 7 K-means iterations, after which the weights and variance vectors are initialized. Finally, two EM iterations are performed.

When adapting the target models, we adapt only the mean vectorsa using a relevance factor $r = 16$. During recognition, the N top scoring Gaussians are found from the UBM for each feature vector, and only the corresponding adapted Gaussians in the target model are evaluated. Match score is the difference of the target model and the UBM log-likelihoods.

The VQ–UBM system (Hautamäki et al., 2008) is similar to GMM–UBM but simpler in its implementation. The UBM consists of only centroid vectors without any variance or weight information. Two gender-dependent UBMs are trained using the splitting method, followed by 20 K-means iterations. The centroids are adapted by using a modified K-means algorithm. We use a relevance factor $r = 12$ and $I = 2$ iterations as in (Hautamäki et al., 2008). In the recognition phase, the N closest UBM vectors are searched for each vector. In the speaker model, nearest neighbor search is limited on the corresponding adapted vectors only. Match score is the difference of the UBM and target quantization errors.

GLDS–SVM system follows the basic implementation presented in (Campbell et al., 2006a). The 36-dimensional MFCCs are first expanded by calculating all the monomials up to order 3, implying a new feature space of 9139 dimensions. The expanded features are then averaged to form a single *characteristic vector* for each utterance. During enrollment, the target speaker vector is labeled as +1, whereas the background vectors are labeled as −1. Similar to GMM–UBM and VQ–UBM, we use gender-dependent background sets also for GLDS–SVM. The characteristic vectors, assigned with appropriate labels, are then used for training the SVM. The commonly available *Statistical Pattern Recognition Toolbox*[1] is used for this purpose. The result of the training is a model vector of dimension 9139. The match score is computed as the inner product between the model vector and the vector corresponding to the test utterance.

In addition to the three base classifiers, we consider their *fusion* by linear score weighting. The scores are standardized using global mean and variance estimates of the scores prior to weighting. We use a logistic regression objective function, as implemented in the FoCal toolkit,[2] to optimize the fusion weights. In preliminary experiments, we experimented with several other solutions such as Bayes nets and neural networks. The logistic regression yielded the best fusion gain on average and was therefore chosen.

2.3. Corpora and performance evaluation

We use NIST 2005 and NIST 2006 speaker recognition evaluation (SRE) data sets for optimizing the parameters, of which the

[1] http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html.
[2] http://niko.brummer.googlepages.com/focal.

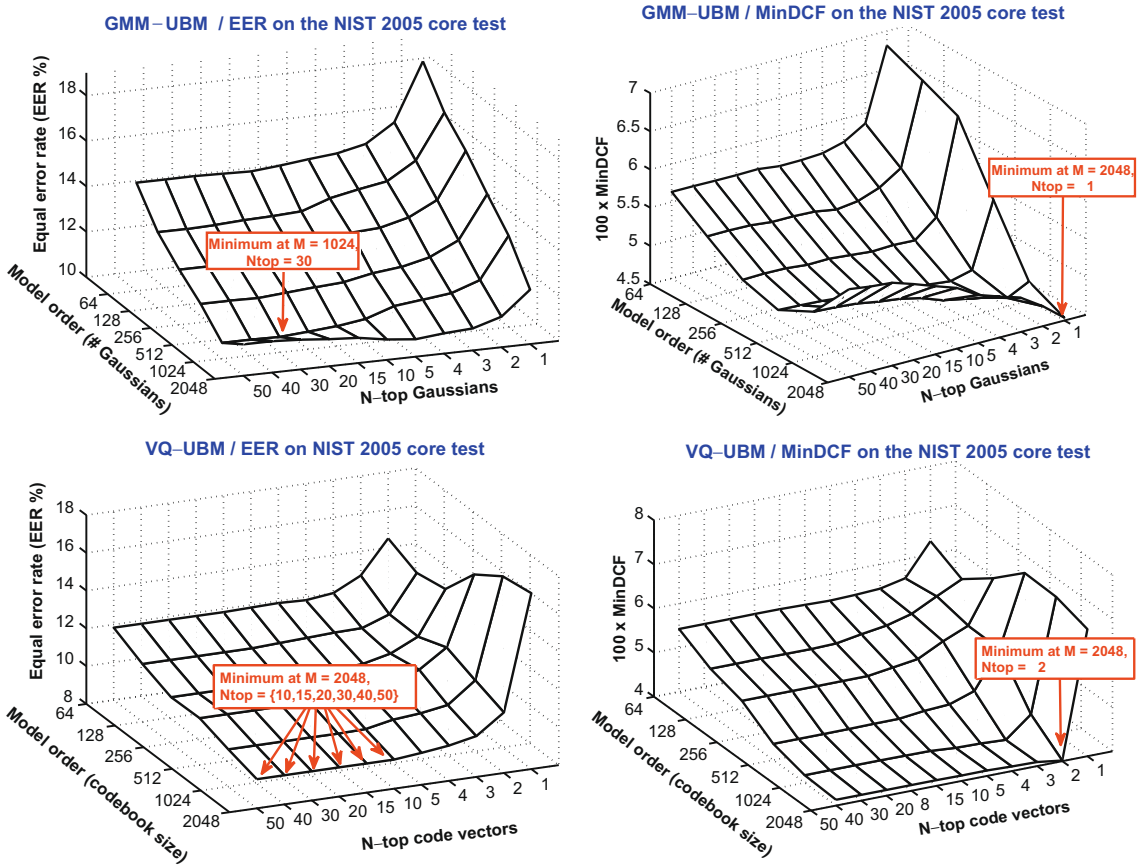

Fig. 1. Effect of *N*-top scoring to recognition accuracy.

most important is the *model order* (number of Gaussians and centroids in GMM and VQ, respectively). Furthermore, we use the latest NIST 2008 SRE corpus to investigate the effect of mismatched data – the 2008 SRE data contains, for instance, interview data that is not present in the other corpuses.

In all three corpora, we focus on two test conditions. The first is the "core" test, referred to as "1conv–1conv" in the NIST 2005/2006 corpora and "Short2–Short3" in the NIST 2008 corpus. It contains 5 min of telephone-quality train and test data of which about half (2.5 min) contains speech. The second test condition known as "10sec–10sec", contains only 10 seconds of training and test data. The 1-conversation training files of the NIST 2004 corpus (246 males and 370 females) are used as the background utterances for all three classifiers. To simplify system optimization and save processing time, we use the same background training set for all three corpora.

In evaluating our recognizer performance, we use two well-known metrics. The first one, *equal error rate* (EER), corresponds to the decision threshold that gives equal false acceptance rate (FAR) and false rejection rate (FRR). The second measure, referred to as *minimum detection cost function* (MinDCF), punishes heavily false acceptances. It is used in the NIST SRE evaluations[3] and defined as the minimum value of the function $0.1 \times \mathrm{FRR} + 0.99 \times \mathrm{FAR}$. The minimum is taken over all possible decision thresholds. For more details about evaluation of speaker verification performance, refer to (Briimmer and Preez, 2006).

[3] http://www.nist.gov/speech/tests/sre.

3. Optimization results: NIST 2005/2006

3.1. *N*-top Scoring

First, we study the number of top scoring Gaussians and code vectors for the GMM–UBM and VQ–UBM systems for match score computation because of several reasons. Firstly, we are not aware of a systematic study on the effect of *N*-top value to the recognition accuracy. In (Reynolds et al., 2000), it is stated that $N = 5$ top scoring components are enough. We hypothesized that for higher model orders, more *N*-top Gaussians would be required for accurate recognition as the likelihood computation gets more accurate. On the other hand, VQ–UBM obtains *exactly the same result* as full search if the nearest code to the unknown vector is in the *N*-top list. This made us hypothesize that VQ–UBM may require a smaller value of *N*.

Fig. 1 displays the EER and MinDCF values for the studied parameter combinations. We make the following immediate observations. First, GMM–UBM is somewhat sensitive to the selection of *N*; the optimum value depends on both the objective function (EER, MinDCF) and the model size. VQ–UBM, on the other hand, is less sensitive to value of *N*; any value $N \geqslant 10$ minimizes both EER and MinDCF. Moreover, the result is fairly independent of the model order. The GMM–UBM and VQ–UBM have some similarities as well. In particular, both models achieve a small EER for "large" *N* and small MinDCF for "small" *N*.

Does larger model require more *N*-top components as we hypothesized? According to the results shown here the answer is ***no***. Even the opposite can happen. For instance, the MinDCF of the GMM–UBM increases with *N* for large model sizes. In other

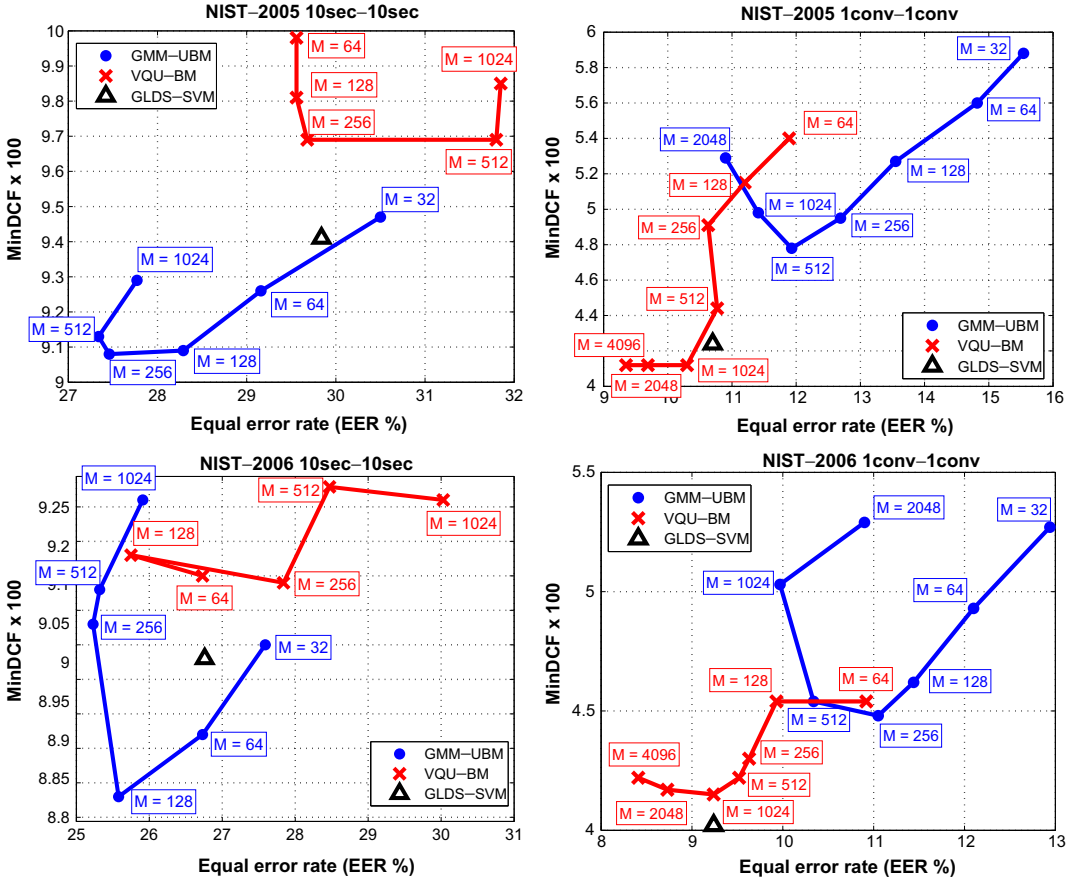

Fig. 2. Accuracy on the NIST 2005 and NIST 2006 corpora for the 10sec–10sec and 1conv–1conv tests.

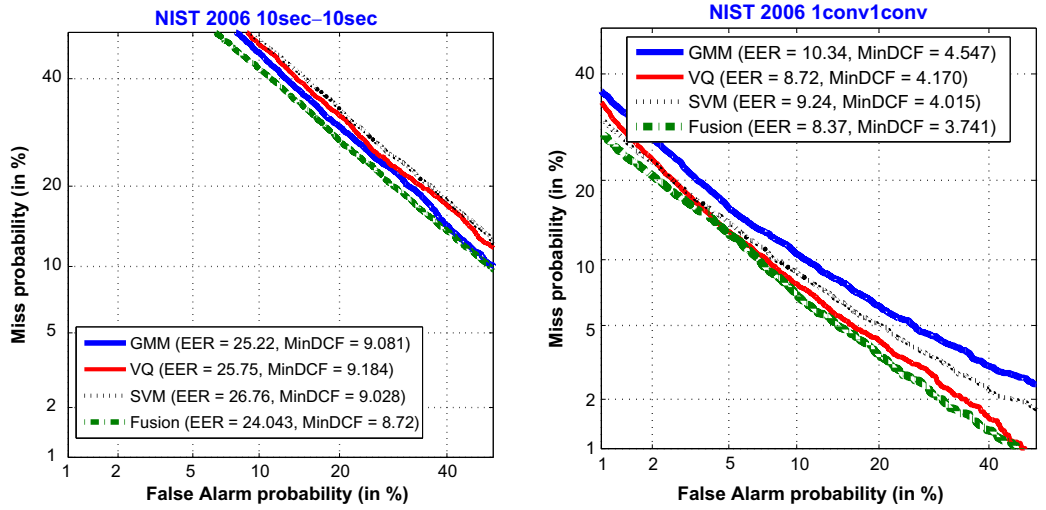

Fig. 3. Fusion results on the NIST 2006 corpus.

words, the more inaccurate the computation of the log-likelihood ratio, the better MinDCF! This is an indirect indication of sub-optimal speaker model density estimation, and possibly some other violations in the modeling assumptions. For the rest of the experiments, we fix the values $N = 10$ for the GMM–UBM and $N = 5$ for the VQ–UBM. These values were chosen so as to yield a small EER with significant speed-up compared to full search.

3.2. Model order

The results for the different classifiers are summarized in Fig. 2 which displays EER against MinDCF. For the GMM–UBM and VQ–UBM classifiers, results are shown for different model orders M. The GLDS–SVM does not have a similar control parameter and hence is presented by a single point.

We make the following observations:

- Optimal model order depends on both the test condition and on the type of the model (GMM–UBM or VQ–UBM); for shorter training and test data the optimal model order is lower compared with longer training and test data.
- GMM–UBM outperforms VQ–UBM for the short training and test condition (10sec–10sec), and vice versa, VQ–UBM outperforms GMM–UBM on the long training and test conditions (1conv–1conv).
- The GMM–UBM performance is consistent across the two data sets giving nicely convex error curves for both corpora and conditions.
- Accuracy of the SVM lies in between the other two classifiers for the 10 s test cases. It gives comparative results to VQ–UBM classifier on the longer test case (1conv–1conv). It also shows consistent (predictable) performance for both test cases.

3.3. Fusion

Next, we study fusion performance on the NIST 2006 corpus. The fusion weights and the score standardization parameters were obtained from the NIST 2005 corpus. The fusion weights were separately optimized for the core and the 10sec–10sec conditions. The results are displayed in Fig. 3. The difference between the best and worst classifiers is less than 2% unit in both cases. The choice of the classifier is therefore not critical for the overall performance assuming that the parameters have been optimized properly. Fusion of the three components provides slight but consistent improvement in comparison to the best individual classifier.

In addition to the slight improvement in accuracy, another benefit of the fusion is to reduce the risk of selecting wrong classifier. Even if an individual classifier works well for one kind of data, it may fail for another. In our tests, fusion seems to avoids this problem. On the other hand, fusion as a secondary classifier in top of the base classifiers complicates the system optimization which is not attractive for practical application.

4. Results on the NIST 2008 corpus

The optimized classifiers were then evaluated on the NIST 2008 data. The results shown here are based on our primary submission system to the NIST 2008 SRE campaign. Due to page limitations, only a few selected cases are shown.

The following model sizes were used for the Short2–Short3 and 10sec–10sec, respectively: 512, 256 (GMM), and 2048, 128 (VQ). The core test contains several subcases with different types of data mismatches. The most challenging are cross comparisons of telephone data and the interview data recorded with different

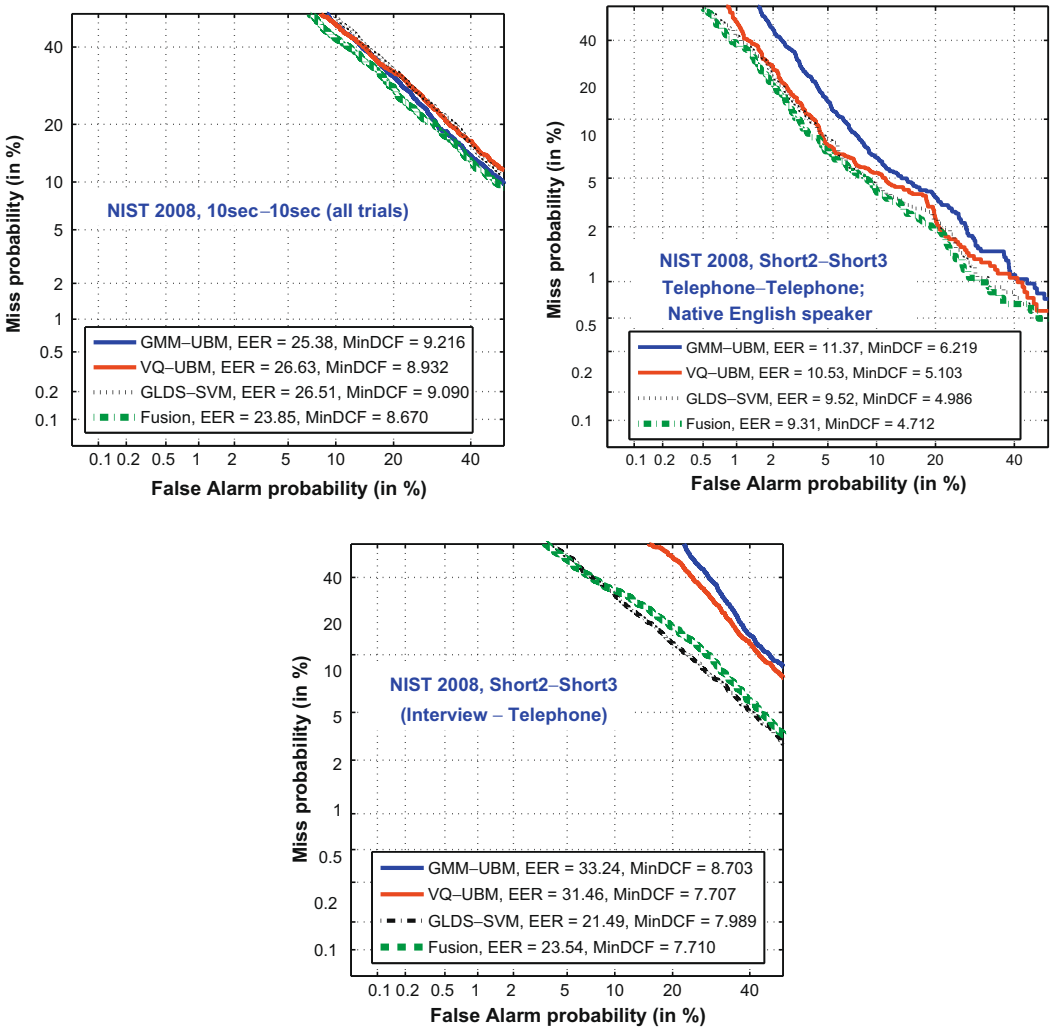

Fig. 4. Selected results on the NIST 2008 corpus.

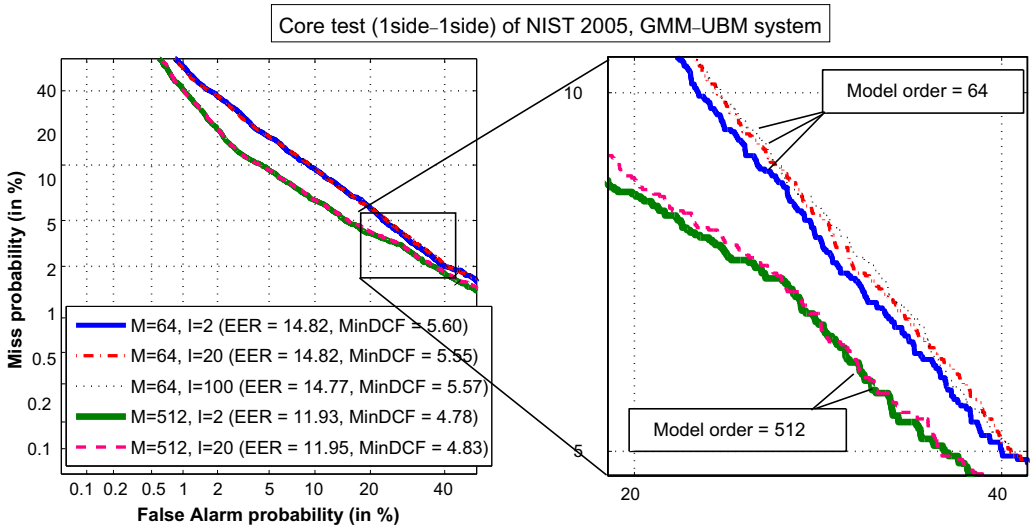

Fig. 5. Effect of the number of EM iterations for UBM training.

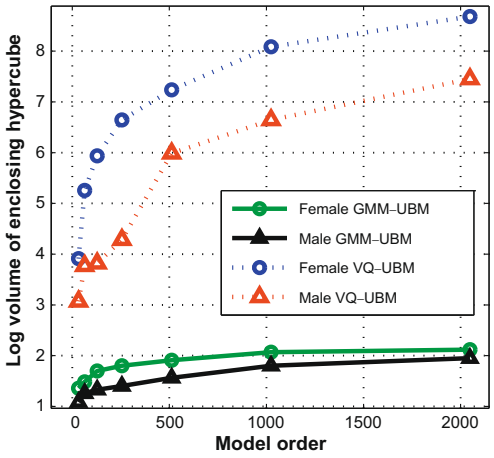

Fig. 6. Volumes of the hypercube enclosing the UBM.

microphones of varying quality. Selected results shown in Fig. 4 are two fold. On one hand, for the 10sec–10sec test case, the observations made for NIST 2006 results generalize well to the 2008 corpus: GMM–UBM is the best individual classifier and fusion gives slight improvement of the accuracy. The results for the Short2–Short3 telephone data are also consistent with those of NIST 2006 1conv–1conv case as GMM–UBM is still the worst. However, GLDS–SVM performs now slightly better than VQ–UBM.

On the other hand, the interview material does not exist in our background training data nor in the NIST 2005 and NIST 2006 evaluation data. The methods tuned for these corpora do therefore not apply well to the trials where interview data is present. The results with the worst channel mismatch (interview–telephone case), GLDS–SVM appears to be most robust.

5. Discussion

The most interesting observation, in our opinion, is that VQ–UBM clearly outperforms GMM–UBM on the longer training and test data, whereas GMM–UBM is better for short training and test samples. This contradicts intuition and our initial hypothesis: since speaker models in the VQ–UBM approach have less parameters, one would expect it to perform better on short samples.

Are the differences between the GMM–UBM and VQ–UBM due to the inherent differences in the models themselves or just because of differences in their parameter settings? One may argue that, as we used only 2 EM iterations to train the background model for the GMM–UBM system and 20 K-means iterations for the VQ–UBM, the setting is unfair for GMM. To study this, we varied the number of EM iterations for the UBM training in the GMM–UBM system as a post-evaluation analysis. The results displayed in Fig. 5 clearly indicates that the number of EM iterations (I) is an insignificant parameter compared to selection of the model order (M). In other words, maximum likelihood criterion training of the UBM is not optimal; if this was the case, further iterations would improve accuracy.

To gain further insight into the structure of the models, we analyzed the volume of the hypercube that encloses the UBM in each model. For the VQ–UBM model, we found the 95-percentile intervals containing any centroid in each dimension. For the GMM–UBM model, we found the 95-percentile intervals containing the $\mu \pm 3\sigma$ points.

The result in Fig. 6 indicates that the background model in the VQ–UBM system is much more spread out in the feature space, yielding possibly more flexible adaptation for longer training and test data. One hypothesis is that the VQ–UBM adaptation may better take into account some infrequently occurring speech sounds, which are considered outliers in the GMM for large amounts of data. The Gaussians in the GMM–UBM seem to concentrate on a denser region. On the other hand, it can be hypothesized that the variance information helps to "interpolate" between the training feature vectors, yielding better generalization performance on limited data tasks.

6. Conclusion

In this paper, we have experimentally compared VQ- and GMM-based speaker models trained using MAP criterion. The most surprising observation was that VQ–UBM gave better results for longer training and test segments whereas GMM–UBM was better for short segments. Analysis of the background models revealed that the UBM in the VQ system is more spread out in the feature space, yielding possibly more flexible adaptation for longer training and

test data. In summary, the differences seem not only due to parameter settings but in the model types themselves.

In future, it would be interesting to artificially de-sharpen the Gaussians in GMM to see if the accuracy gets better. It would be also interesting to study the combination of the VQ–UBM and support vector machine as already done for the GMM–UBM by several authors (Lee et al., 2008; Campbell et al., 2006b). Finally, an important point is to improve robustness against session variability; for this, adopting the recently proposed eigenchannel or JFA compensation techniques (Burget et al., 2007; Kenny et al., 2008; Vogt and Sridharan, 2008) for the VQ–UBM model seems a promising direction.

References

Briimmer, N., Preez, J., 2006. Application-independent evaluation of speaker detection. Comput. Speech Lang. 20, 230–275.

Burget, L., Matejka, P., Schwarz, P., Glembek, O., Cernocky, J., 2007. Analysis of feature extraction and channel compensation in a GMM speaker recognition system. IEEE Trans. Audio, Speech. Lang. Process. 15 (7), 1979–1986.

Campbell, W., Campbell, J., Reynolds, D., Singer, E., Torres-Carrasquillo, P., 2006a. Support vector machines for speaker and language recognition. Comput. Speech Lang. 20 (2–3), 210–229.

Campbell, W., Sturim, D., Reynolds, D., 2006b. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Process. Lett. 13 (5), 308–311.

David, P., 2002. Experiments with speaker recognition using GMM. In: Proc. Radioelek-tronika 2002. Bratislava, pp. 353–357.

Hautamäki, V., Kinnunen, T., Kärkkäinen, I., Tuononen, M., Saastamoinen, J., Fränti, P., 2008. Maximum a posteriori estimation of the centroid model for speaker verification. IEEE Signal Process. Lett. 15, 162–165.

He, J., Liu, L., Palm, G., 1999. A discriminative training algorithm for VQ-based speaker identification. IEEE Trans. Speech Audio Process. 7 (3), 353–356.

Hermansky, H., Morgan, N., 1994. RASTA processing of speech. IEEE Trans. Speech Audio Process. 2 (4), 578–589.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Du-mouchel, P., 2008. A study of inter-speaker variability in speaker verification. IEEE Trans. Audio, Speech Lang. Process. 16 (5), 980–988.

Lee, K., You, C., Li, H., Kinnunen, T., Zhu, D., 2008. Characterizing speech utterances for speaker verification with sequence kernel SVM. In: Proc. 9th Interspeech (Interspeech 2008). Brisbane, Australia, pp. 1397–1400.

Reynolds, D., Campbell, W., Gleason, T., Quillen, C., Sturim, D., Torres-Carrasquillo, P., Adami, A., 2005. The 2004 MIT Lincoln laboratory speaker recognition system. In: Proc. Internat. Conf. on Acoustics, Speech, and Signal Process. (ICASSP 2005), vol. 1. Philadelphia, USA, pp. 177–180.

Reynolds, D., Quatieri, T., Dunn, R., 2000. Speaker verification using adapted gaussian mixture models. Digit. Signal Process. 10 (1), 19–41.

Soong, F.K., Rosenberg, A.E., Juang, B.-H., Rabiner, L.R., 1987. A vector quantization approach to speaker recognition. AT&T Tech. J. 66, 14–26.

Tong, R., Ma, B., Lee, K., You, C., Zhu, D., Kinnunen, T., Sun, H., Dong, M., Chng, E., Li, H., 2006. Fusion of acoustic and tokenization features for speaker recognition. In: 5th Internat. Symp. Chinese Spoken Lang. Process. (ISCSLP 2006). Singapore, pp. 494–505.

Vogt, R., Sridharan, S., 2008. Explicit modeling of session variability for speaker verification. Comput. Speech Lang. 22 (1), 17–38.