2001: A Speaker Odyssey
The Speaker Recognition Workshop
Crete, Greece
June 18-22, 2001

Gaussian Selection Applied to Text-Independent Speaker Verification

Roland Auckenthaler and John S. Mason

Department of Electrical & Electronic Engineering,
University of Wales Swansea, SA2 8PP, UK
email: rauckenthaler@hotmail.com, J.S.D.Mason@swansea.ac.uk

Abstract

Fast speaker verification systems can be realised by reducing the computation associated with searching of mixture components within the statistical model such as a Gaussian mixture model, GMM. Several improvements regarding computational efficiency have already been proposed for speaker verification.

In this paper, the technique of Gaussian selection is applied to the speaker verification task. Gaussian selection is commonly known in speech recognition where it is used to speed up the scoring process of HMM recognisers. Here we use the same technique to reduce the computation of mixture components within the GMM framework. Experiments compare different selection methods on the text-independent Odyssey 2001 speaker verification database. Further, the selection methods are compared with the baseline approach of scoring all mixture components in the full model. The results reveal a computational reduction of factor ten with only minor degradation in verification performance.

1. Introduction

With emerging applications for speaker verification, such as home shopping, the computational efficiency becomes more important. At peak times, these systems may perform several verifications in parallel and hence tend to be overloaded or use large server farms to prevent overloading. To minimise the cost of hardware it might be beneficial to reduce the computational effort for verification. Although the aforementioned verification systems may use a text-dependent mode, similar computational reduction may apply to a text-independent mode.

Most of today's text-independent speaker verification systems model the acoustic space using Gaussian mixture models [1]. These systems use a world model to overcome shortages in speaker training data. A speaker specific model is created from the world model by adaptation. The described speaker verification system needs a large amount of computation for scoring Gaussian densities. With no optimisations, such a system tends to use more than 90% of the processing time for scoring Gaussian densities. An adaptive speaker verification system already reduces the processing by using a mixture correspondence between the world and speaker model [1]. It is known that a good scoring Gaussian density of the world model will lead to a high likelihood for the corresponding speaker model density due to the adaptation. This correspondence leads to scoring only a small number of components for the speaker model and hence reduces the Gaussian density processing by almost a half.

Given the speaker-world correspondence, the processing requirements of an adaptive speaker verification system are related to three factors. These are the dimension of the feature vector, the number of frames per second and the number of components in the world model. Applying linear discriminant analysis [2] or similar techniques [3] have already been proven successful in reducing the dimension of the acoustic space. Experiments on selecting certain frames of the continuous stream for scoring [4] or down-sampling of the signal [5, 6] have been applied to reduce the frame rate for processing. These techniques achieve a processing reduction of about factor 4 with minimal degradation in verification performance. The third factor, reducing the number of components in the world model leads to certain degradation in performance due to a less detailed description of the acoustic space.

In speech recognition, reducing the number of components is performed by processing only a sub-set of mixture components for each frame. In the case of GMMs, this could be achieved by selectively calculating only those components, which are likely to obtain good scores. One technique that reduces processing in speech recognition is called Gaussian selection [7, 8]. Here smaller models, further referred to as hash models, are computed. For each component of this hash model a shortlist of indices is generated which contains indices of mixture components in a larger model which is used for verification. In this paper, different techniques of Gaussian selection are investigated in the context of text-independent speaker verification.

2. Gaussian Selection

A basic text-independent speaker verification system is based on a large Gaussian mixture model, which represents the world. Such a model is trained using an expectation maximisation algorithm [9]. This large base GMM is the fundamental model for scoring the feature vectors and for verification. For Gaussian selection, a smaller, hash model is created which contains shortlists of possibly good scoring mixture component in the larger base GMM. The hash GMM can be trained using different approaches. Some of them are discussed later.

The hash model is the first stage in scoring. A feature vector is first scored against all components in the hash model. The best scoring component is then used to retrieve indices of components in the larger base GMM. These indices are stored in a shortlist for each particular mixture component and are obtained during a separate training phase.

2.1. GMM Hashing

The most straightforward method of creating a hash model is to train a smaller GMM with the same training data as the large base GMM. Given a base model size of 1024 components, a hash model with 32 components can index all components in the large model when at least 32 indices are stored in each shortlist. This would lead to an overall scoring of 64 mixture components for each feature vector instead of the full model processing of 1024 component.

After creating the hash model, the shortlists are generated by a separate step where each of the training feature vectors is scored against the hash model and the large base model. This leads to an index pair of best scoring components for each feature vector. These index pairs are obtained for each feature vector and a histogram of occurrences is calculated. A shortlist of indices to components in the large base model is obtained by sorting the occurrence frequencies. This is performed for each individual hash component separately.

The shortlists allow predicting the most likely indices in the base model. The number of components to process can vary from a few to some hundreds depending on the performance degradation, which can be accepted for reducing the processing. In the case of 1024 components in the base model and 32 components in the hash model, the critical shortlist size is 32. This would at least allow the possibility of indexing all components in the large base model when each component index occurs exactly once. A smaller shortlist will lead to certain components in the base model, which will never be processed and hence could be discarded from the overall system. Larger shortlist sizes allow the use of certain component indices more than once and hence lead to a more accurate prediction.

The second approach examined in the paper for creating shortlists is similar to the first approach. The difference is that each component index is associated with exactly one component in the hash model. This is done by successively assigning the most likely index to a hash component. These likelihoods are given by the frequencies of occurrences as index pairs during training. If an index is already assigned to a component, it is discarded for further use in shortlists of other components.

The third approach investigated in this paper is the original approach of [7]. Here the hash model is created using the mixture components mean vectors of the large base GMM. The mean vectors are the training data for the hash model of 32 mixture components. After training the hash model, each base model mean vector is scored against the hash model. This leads to a best scoring hash component. The base model component index is then inserted into the shortlist of this particular hash component.

The generation of shortlists reveals that only about 500 different components are indexed for a certain hash component. The others are never observed in a particular component index pair. This indicates that the processing can be reduced by factor two without any loss in performance.

The use of the second approach, further referred to as GS1, reveals that the shortlist sizes are fairly balanced with 23 and 45 indices per list. The third approach, further referred to as GS2, obtained a far larger variation in the size of the shortlists. Here the shortlist size varied between 16 and 98. This seems quite large and might cause some performance degradation because a large amount of base model indices are associated to a small number of hash components. In the original approach, using a so-called gauge factor circumvents possible performance degradations due to an imbalance in the list sizes. This factor allows a base model component index to be assigned to more than just one hash model component. Therefore the performance degradation is compromised against more processing.

3. Experimental Setting

The speaker verification system uses a front-end frame rate of 62.5 frames per second. An FFT is applied to the signal frames with no overlap between frames. The spectral information is down-sampled to 16 linear spaced energy bins and a logarithmic compression is applied. The spectral features are transformed to the cepstral domain where a channel normalisation is applied using RASTA filtering [10]. Further, first order derivatives are calculated and appended to the feature vector. These transformations obtain an acoustic space with 32 dimensions.

The GMM system uses a world model with 1024 mixture components. This model is trained using a database of British English. Recordings of both genders are used to maximise the amount of training data available, which are recorded using different handsets and telephone speech quality. From this world model the speaker model is created by adapting mean parameters only. As a final step in scoring, a simple world normalisation [11] is applied.

The target speaker set comprises the text-independent

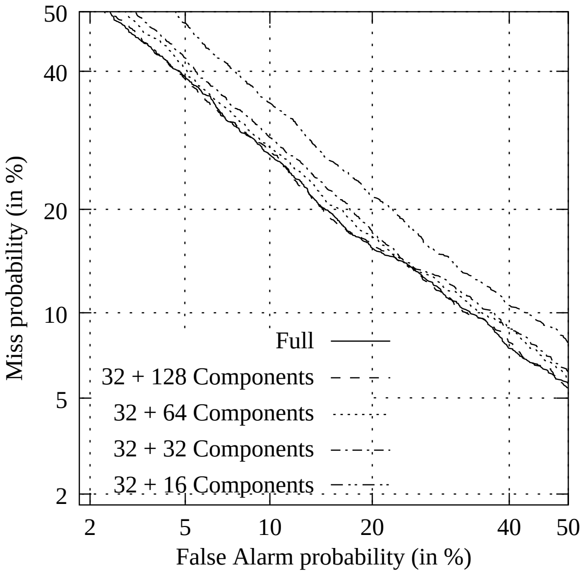

Figure 1: Speaker Verification Performance for Different Shortlist Sizes

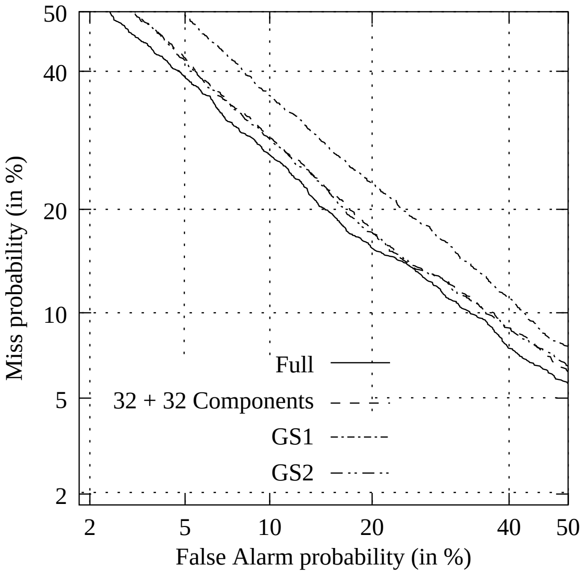

Figure 2: Comparison of Different Training Approaches for the Hash Model and Shortlists

part of the Odyssey 2001 evaluation data. These data are the male speaker set from the NIST 2000 evaluation and contain two minutes of training data for each speaker. The test utterances vary in length form 15 to 45 seconds and are recorded from a different phone number (different handset) to the training handset of the target speaker. The evaluation comprises the use of electret handset only.

4. Results

The first set of experiments examines the variation of shortlist sizes for the first hashing approach. The shortlist size is varied between 16, 32, 64 and 128 indices and compared to the processing of the full base model.

Figure 1 shows the performance degradations when the shortlist size is varied. The plots reveal no degradation when only 128 components are calculated for the large base model. A further reduction of the shortlist size down to 32 components reveals a small degradation. Reducing the shortlists even further, such as down to 16 components, reveals large degradations in verification performance. This indicates that the critical list size of 32 components might be a lower limit for a compromise between computational reduction and increase in verification error. Using a shortlist size of 32 components leads to processing a total of 64 mixture components for each feature vector. This equals a computational reduction of factor 16. Processing a total of about 100 components, 32 hash components and 64 base components, still leads to only small reductions in performance. Here the reduction in processing is of about a factor of 10.

The second set of experiments compares the processing of the full models with the different training strategies for the shortlists. Here the result of the full processing is compared against the standard shortlist training method with a list size of 32 components. Further the GS1 and GS2 training methods are included in the comparison.

Figure 2 reveals a significant degradation when the GS2 training method is used to generate the shortlists. The main cause for the degradation might be the imbalance in the shortlist sizes, which may lead to a poor indexing of components from the large base model. As mentioned previously this can be avoided by using a gauge factor with the side effect of more processing. The GS1 indexing reveals similar performance to the basic prediction with an equal shortlist size of 32 indices across all hash model components.

5. Discussion

So far only simple approaches have been applied to the reduction of processing. Another approach to reducing the processing would be to train a hierarchical tree structured GMM [12] with more or less levels. This allows dividing the training data in each hierarchic level and thus reduces the likelihood of miss-indexed components in the large base model. Therefore the performance might not degrade or even improve slightly due to a more component specific training.

A tree structured GMM might also be advantageous when the speaker training data are limited. The ”tree-model” can grow with additional speaker training data and hence improve the performance. Further the use of a tree structure allows introducing new normalisation techniques which make use of the different levels of detail in the acoustic representation in the tree.

Another issue of applying Gaussian selection is the additional memory overhead. This might be important for embedded applications where memory resources are limited. Given the scenario of a base model size with 1024 components, a hash model size of 32 components and shortlists with 32 indices each, the memory overhead due to Gaussian selection is less than 4%. This seems a good compromise for a processing reduction of factor 16. The overhead may not be important because the additional parameters can reside in ROM address space. The speaker model on the other hand resides in more costly RAM or EEPROM.

6. Conclusions

In this paper the approach of Gaussian selection is applied to the task of text-independent speaker verification. Gaussian selection allows to reduce the processing of Gaussian mixture components in the verification system by creating a smaller hash model and use this to index likely mixture components in the larger base model. This form of reducing the processing is useful when a large number of verifications are performed in a very short period.

The approaches discussed in this paper create a small model of 32 components to index the large base model of components. The shortlists of indices are stored for each hash model component individually, which allows to vary the number of base model indices. Different approaches have been investigated to create the index tables. The most favourable approach trains the most likely indices of each hash model component by scoring each training vector against the hash model and the large base model. The generated index pairs are sorted by their occurrence frequencies and the most likely indices are kept for each hash model component. This allows to trade off verification performance against a computational reduction.

The experiments reveal that scoring only 128 out of 1024 mixture components of the large base model leads to no noticeable performance degradation. This is a processing reduction of about factor 6. A further reduction down to 32 processed components for the base model only leads to minor degradation in verification performance. Here the computational reduction of factor 16 is achieved with slight degradations in verification performance and a less than 4% memory overhead. Reducing the computation even further reveals large decreases in performance.

7. References

[1] D. Reynolds and T. Quatieri. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing, Academic Press*, 10(1-3):19–41, 2000.

[2] M. Hunt. An Investigation of PLP and IMELDA Acoustic Representations and of Their Potential for Combinsation. In *Proc. ICASSP '91*, volume 2, pages 881–884, Toronto, Canada, 1991.

[3] H. Hermansky and N. Malayath. Speaker Verification using Speaker-Specific Mappings. In *Proc. RLA2C '98*, pages 111–114, Avignon, France, 1998.

[4] J. McLaughlin, D. Reynolds, and T. Gleason. A Study of Computation Speed-Ups of the GMM-UBM Speaker Recognition System. In *Proc. Eurospeech '99*, volume 3, pages 1215–1218, Budapest, Hungary, 1999.

[5] S. van Vuuren and H. Hermansky. !MESS: Modular, Efficient Speaker Verification System. In *Proc. RLA2C '98*, pages 198–201, Avignon, France, 1998.

[6] S. van Vuuren and H. Hermansky. On the Importance of Components of the Modulation Spectrum for Speaker Verification. In *Proc. ICSLP '98*, Sydney, Australia, 1998.

[7] E. Bocchieri. Vector Quantization for the Efficient Computation of Continuous Density Likelihoods. In *Proc. ICASSP '93*, pages 692–695, Minneapolis, 1993.

[8] M. Gales, K. Knill, and S. Young. State-Based Gaussian Selection In Large Vocabulary Continuous Speech Recognition Using HMMs. *IEEE Trans. Speech and Audio Processing*, 7(2):152–161, 1999.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, 39:1–38, 1977.

[10] H. Hermansky. Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP). In *Proc. EuroSpeech '91*, volume 3, pages 1367–1370, 1991.

[11] M. Carey, E. Parris, and J. Bridle. A Speaker Verification System Using Alpha Nets. In *Proc. ICASSP '91*, volume 1, pages 397–400, 1991.

[12] S. Maes U. Chaudhari, J. Navratil and R. Gopinath. Transformation Enhanced Multi-Grained Modelling for Text-Independent Speaker Recognition. In *Proc. ICSLP '00*, Beijing, China, 2000.