Multi-level Gaussian selection for accurate low-resource ASR systems

Leïla Zouari, Gérard Chollet
GET-ENST/CNRS-LTCI
46 rue Barrault, 75634 Paris cedex 13, France

Abstract

For Automatic Speech Recognition ASR systems using continuous Hidden Markov Models (HMMs), the computation of the state likelihood is one of the most time consuming parts. As the performance and the speed of ASR systems are closely related to the number of HMM Gaussians, removing Gaussians without decreasing the performance is of major interest. Hence, we propose a novel multi-level Gaussian selection technique to reduce the cost of likelihood computation. The global process starts from an accurate system containing large Gaussian mixtures. Gaussian distributions are then organized into a hierarchical structure (ie. tree) and multiple classifications are performed by cutting the tree at different levels. While traversing the tree through the levels of cut only likely nodes are kept. Later, a data-based pruning procedure is applied to the selected components.

An hour of Ester, a French broadcast news database is used for the test. The experiments that show that increasing the number of levels is advantageous and using the data-based pruning procedure reduces the number of computed densities without a significant decrease in system performance.

Keywords: Speech, recognition, Gaussian, selection, likelihood.

1. Introduction

The proliferation of mobile devices in daily life has created a great demand for efficient and simple interfaces to these devices. In particular, speech recognition being a key element of the conversational interface, there is a significant requirement for low-resource, robust and accurate speech recognition systems.

Recent mobile devices offer a large set of functionalities but their resources are too limited for accurate continuous speech recognition engines. Indeed, state-of-the-art continuous speech recognition systems use hidden Markov models with many tens of thousands of Gaussian distributions to achieve improved recognition. As the acoustic matching often occupies most of the decoding time and that only a few Gaussians dominate the likelihood of a Gaussian mixture different techniques were developed to select them.

Bocchieri (1993) proposed a Gaussian selection technique by vector quantization. It generates a vector quantized codebook and attributes a shortlist of Gaussians to each codebook entry. During decoding, the frame is assigned to the nearest

codebook. Gaussian distributions belonging to the corresponding shortlist contribute to that frame likelihood computation. An extension to this work (Knill et al 1999) consists in applying a constraint to the number of Gaussians belonging simultaneously to the same state and shortlist. For a LVCSR task, this leads to a decrease in the acoustic matching cost by a factor of six.

In this paper, we propose an original Gaussian selection technique that aims to improve simultaneously the classification and the selection processes. The first objective is satisfied by means of a multi-level Gaussian classification and the use of the symmetric and weighted Kullback-Leibler distance, whose efficiency has been proven in (Zouari et al 2006). The selection is enhanced becauce it is applied to several levels. The overall proposed algorithms can be summarized in two steps : The first one consists in organizing Gaussian distributions belonging to the same state into a binary tree structure. Codewords are the nodes (Gaussian distributions) obtained by cutting the tree at a specified level. Several cuts can be performed. In the second step (ie. selection) the codeword likelihood is computed and sorted. Only the most likely codewords are considered when going on to the lower level of cut. When the leaves of the tree are reached, the corresponding Gaussian distributions are sorted by weight. Finally, only Gaussian distributions having the highest weights are selected for the likelihood computation.

The outline of the rest of the paper is as follows. In section 2, the proposed method with its two steps of classification and selection is described. In section 3, test protocols and experiments are depicted and then the results are commented on. Finally, in section 4, we conclude the paper and propose some perspectives.

2. Hierarchical clustering and selection

The Gaussian selection algorithm is performed in two steps : clustering/ classification and selection. During the first step, the Gaussian distributions of each mixture are grouped into a binary tree and many classifications are obtained by cutting the tree at different levels. The second step consists in selecting distributions to be used for the likelihood computation.

2.1. Gaussian clustering

The clustering algorithm proceeds as follows :

1. Compute the symmetric and weighted Kullback-Leibler distances KLP between all the distributions. If $g_1(n_1,\mu_1,\sigma_1)$ and $g_2(n_2,\mu_2,\sigma_2)$ are Gaussian distributions to which n_1 and n_2 frames have been associated during training, then:

$$KLP(g_1,g_2)=\frac{1}{2}tr(n_1\Sigma_1\Sigma_2^{-1}+n_2\Sigma_2\Sigma_1^{-1})+\frac{1}{2}(\mu_1-\mu_2)^T(n_1\Sigma_1^{-1}+n_2\Sigma_2^{-1})(\mu_1-\mu_2)-(n_1+n_2)d$$

   where d is the dimension of the parameters vectors.

2. Merge the closest distributions. If g_1 and g_2 are merged to $g_3(n_3=n_1+n_2,\mu_3,\sigma_3)$ then:

$$\mu_3=\frac{n_1}{n_3}\mu_1+\frac{n_2}{n_3}\mu_2$$

$$\Sigma_3=\frac{n_1}{n_3}\Sigma_1+\frac{n_2}{n_3}\Sigma_2+\frac{n_2n_1}{n_3^2}(\mu_1-\mu_2)(\mu_1-\mu_2)^T$$

3. If the number of Gaussians is greater than one go to step1.

2.2. Classification

The clustering process organizes the Gaussian distributions into a tree structure. Codewords are the nodes (Gaussian distributions) resulting from cutting the tree at a specified level. A shortlist is assigned to each codeword.

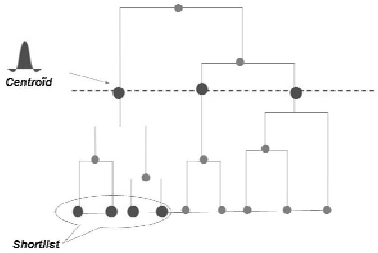

Figure 1 . Gaussians classification

2.3. Multi-criterion selection

While classification is applied before, selection is applied during the decoding process. It aims at detecting, for each node of the decoding graph, Gaussians that dominate the likelihood computation. It operates as follows :

1. Likelihoods of the codewords of the current level of cut are computed. Then they are sorted and the most likely of them are kept before going down to the lower level of cut.
2. When reaching the last level of cut 2 sets of Gaussian distributions can be selected for the likelihood computation :
    a) leaves whose ancestors have all been kept.
    b) leaves selected in a) with large weight values.

The following example illustrates an application of this algorithm to a mixture of 24 Gaussian distributions. Two levels of cut are considered : level 1 and level 2.

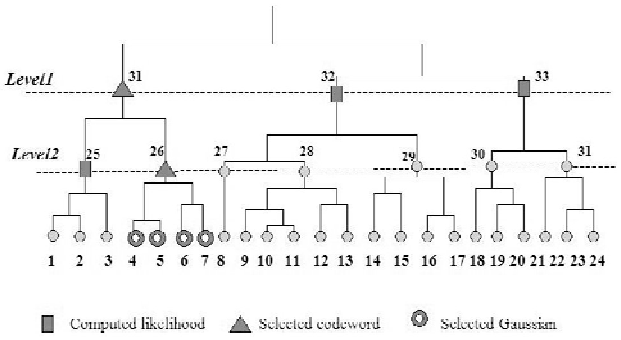

Figure 2 . Example of Gaussian clustering process

First, likelihoods of the codewords 31, 32 and 33 are computed and sorted. As the codeword 31 is the most likely it is selected. Then we move to the next level of cut (level

2) and compute the likelihood of the corresponding nodes that are 25 and 26. If codewords 26 is the likest, the corresponding leaves which are the Gausians 4,5, 6 and 7 are selected. Finally we can decide to compute the likelihood with all of them or to keep only those with the highest weight values.
Hence we computed a total of 9 likelihoods which is less time consuming than 24.

3. Experiments and results

The Sirocco-Htk large vocabulary speech recognition system was used to compare the performance of the different schemes. This reference system uses 40 context independent acoustic models, 3 states each, and 512 Gaussians components per state. The parameter vectors are composed of 12 MFCC coefficients, energy, and their first and second derivatives. For the training task, about 82 manually transcribed hours of the Ester train database (Galleano et al 2005) are used. The dictionary contains 65000 distinct words and the language model is trigram. Tests are conducted using an hour of Broadcast News extracted from the Ester test data set. The Word Error Rate of this reference system is WER = 35.5%.
The performance of the various experiments is addressed in terms of WER and percentage of likelihood computation C. The latter is defined as:
C = computed-likelihoods /all-likelihoods

3.1. One-level based selection

For each state, the 512 Gaussian distributions are organized into a tree structure. Codewords are obtained by cutting the binary tree at a single level. We experimented cutting the tree at the levels 40 and 120 which correspond respectively to 40 and 120 codewords.

- ***Shortlist scores*** : We vary the number of selected codewords and use all the corresponding tree leaves for the likelihood computation. For each number, the corresponding Gaussians and WER are reported in figure 1 (left). As the number of selected Gaussians per state is variable, a mean value is considered. The fraction C is also computed and depicted in Figure1 (right).

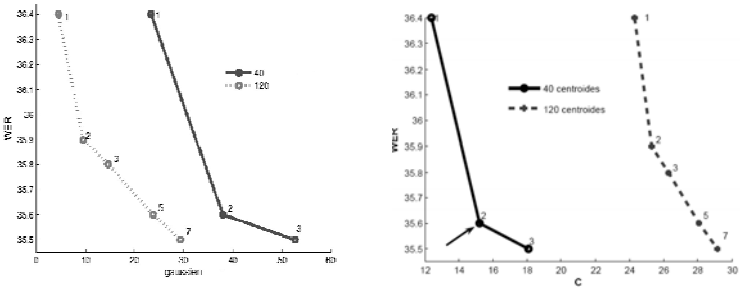

Figure 3. Computing the likelihood using the best shortlists

For the same WER, the number of selected Gaussians is lower when using 120 codewords than 40 codewords. In particular, with only 23 Gaussians per state (Figure 3 left) we obtain exactly the same results as the reference system (512 Gaussians par state). The same experiments (Figure 3 rightb)) show that the value of C is lower for 40

codewords. This is because this fraction takes into account the codebook size.
The best tradeoff between C and the WER corresponds to the selection of 2 codewords and the pair of values (C,WER)=(15.16%,35.6%). In this case the WER increases by only 0.1% and the likelihood computation cost is reduced by a factor of seven.

- ***Data-based selection*** : We take the best system of the previous experiments : 40 codewords among which the 2 likeliest are selected. As the training process is based on the "Maximum Likelihood" criterion, the likely distributions have large weight values. So, to reduce further the number of selected Gaussians, they are sorted by weight and only the components with highest weights are kept. When varying the number of selected Gaussians, we obtain the results of Figure 4.

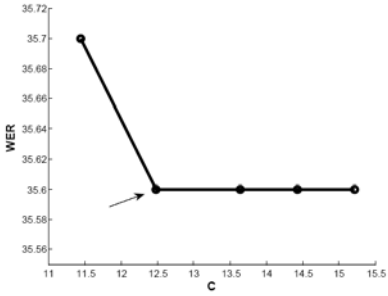

Figure 4. Computing likelihood using the highest weight Gaussians

The best tradeoff between C and WER is (12.48%,35,6%). These results are better than those of the previous experiments. Indeed, for the same value of WER C is reduced. In this case, the likelihood computational cost is decreased by a factor of eight.

3.2. Bi-level selection

The procedure described before is applied by cutting the tree at two different levels. Two bi-levels of cut are experimented : 40-60 codewords and 40-120 codewords.

- ***Shortlist scores*** : In order to improve the results of the previous experiments, all densities of level 40 are computed and the two best codewords are selected. Then we move on to the second level of cut (that is 60 or 120). The corresponding codewords are computed and the most likely of them are kept. Finally, the Gaussians for there codewords are used for the likelihood computation.

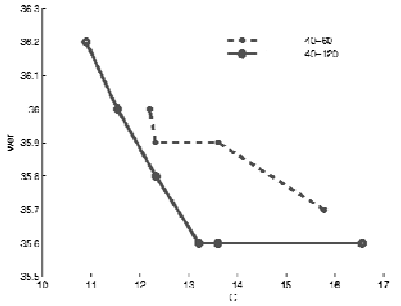

Figure 5. Computing likelihood using two levels of cut and the best shortlists

The following comments may be made :
- the 40-120 curve gives better results than the 40-60 curve. This is foreseeable because the level 120 is lower than the level 60 so the classification is more precise.
- the best tradeoff between C and WER corresponds to the pair of values (C,WER) = (13,21%,35.6%). This result is better than that where the tree was cut at a single level but less good than the result using weight values.

- ***Data-based selection*** : We proceed in the same manner as in the previous experiments, the best settings are considered : bi-level of cut 40-120, and the best pair of values (C,WER)= (13,21%,35.6%). The optimisation of the system consists in keeping only the Gaussians with the highest weight values.
  When varying the number of selected Gaussians, we obtain the results of Figure6.

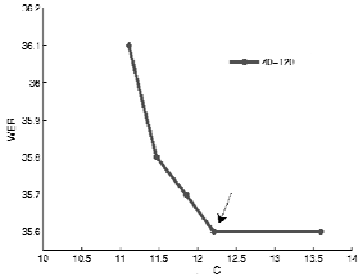

Figure 6. Computing likelihood using the highest-weight Gaussians

Figure 6 shows that from the point (C,WER)= (35,6%, 12.20%), the WER increases. So this point is considered as the best trade off between C and WER.
As the width of the confidence interval is about 0.8\%, other tradeoffs are also satisfactory. It is for example the case of the pair of values (C,WER)= (35,8%, 11.5%) which corresponds to a decrease in the likelihood computation cost by a factor of nine with a non significant loss of accuracy (+0.3%).

4. Conclusion

This paper presents an algorithm to reduce the computation cost in low-resource and large application mobile devices. It consists of a multi-level and robust Gaussian selection method that aims at enhancing simultaneously the classification and the selection processes. The multi-level classification is based on a weighted symmetric Kullback-Leibler distance. The selection is performed at different levels and takes into account the likelihood and the weights of Gaussian distributions.

Experiments on the Ester broadcast news database show that increasing the number of levels and considering the Gaussian weight values reduce the likelihood computation by a factor of 9.

5. Acknowledgments

The authors would like to thank Peter Weyer-brown for his help.

6. Bibliography

[1] Bocchieri, Enrico 1993. Vector Quantization for the efficient computation of continuous density likelihoods. In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Minneapolis. 692–695.

[2] Gales, M.ark; Mc. Knill, Katerine; Young, Steve 1999. State based Gaussian selection in large vocabulary continuous speech recognition using HMMs. In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[3] Zouari, Leila; Chollet, Gérard 2006. Efficient mixture for speech recognition. In : *International Conference in Pattern Recognition, ICPR* , Hongkong, 294–297.

[4] Bonastre, Jean François; Galliano, Sylvain; Geoffrois, Edouard; Mostefa, Djemal; Choukri, Kahlid; Gravier, Guillaume 2005. The Ester Phase II campaign for the rich transcription of French broadcast news. In: *Interspeech.* Lisboa.

LEILA ZOUARI received the electrical engineering diploma from the Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunisia in 1997. She pursue my studies in signal processing and received the DEA degree in 1999 from ENIT, and the Ph.D. degree from the Ecole Nationale Superieure des Telecommunications (ENST), Paris, France. Her Ph.D. research focued on real time large vocabulary speech recognition. She studied several techniques for fast likelihood computation. Then she participated to many projects concerning audio-visual speech recognition and silent interface communication. She is actually at a post-doctoral position in ENST. E-mail: zouari@enst.fr

GERARD CHOLLET Until the doctoral level, his education was centered on Mathematics (DUES-MP), Physics (Maitrise), Engineering and Computer Sciences (DEA). He studied Linguistics, Electrical Engineering and Computer Science at the University of California, Santa Barbara where he was granted a PhD in Computer Science and Linguistics. He taught courses in Phonetics, Speech processing and Psycholinguistic in the Speech and Hearing department at Memphis State University in 1976-1977. Then, he had a dual affiliation with the Computer Science and Speech departments at the University of Florida in 1977-1978. He joined CNRS (the french public research agency) in 1978 at the Institut d Phonétique in Aix en Provence. In 1981, he was asked to take in charge the speech research group of Alcatel. In 1983, he joined a newly created CNRS research unit at ENST where he was head of the speech group. In 1992, he participated to the development of IDIAP, a new research laboratory of the `Fondation Dalle Molle' in Martigny, Switzerland. Since 1996, he is back, full time at ENST, managing research projects and supervising doctoral work. His main research interests are in phonetics, automatic speech processing, speech dialog systems, multimedia, pattern recognition, digital signal processing, speech pathology, speech training aids, ...E-mail: chollet@enst.fr