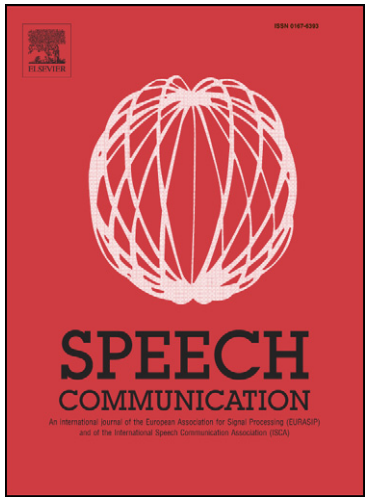

Efficient Codebooks for Fast and Accurate Low Resource ASR Systems

Leila Zouari, Gérard Chollet

PII: S0167-6393(09)00006-5
DOI: 10.1016/j.specom.2009.01.010
Reference: SPECOM 1775

To appear in: *Speech Communication*

Received Date: 30 November 2007
Revised Date: 15 January 2009
Accepted Date: 21 January 2009

Please cite this article as: Zouari, L., Chollet, G., Efficient Codebooks for Fast and Accurate Low Resource ASR Systems, *Speech Communication* (2009), doi: 10.1016/j.specom.2009.01.010

Efficient Codebooks for Fast and Accurate Low Resource ASR Systems

Leila Zouari, Gérard Chollet

GET - ENST / CNRS-LTCI
Département Traitement du Signal et des Images,
46 rue Barrault, 75634 Paris, France.

Abstract

Today, speech interfaces have become widely employed in mobile devices, thus recognition speed and resource consumption are becoming new metrics of Automatic Speech Recognition (ASR) performance.
For ASR systems using continuous Hidden Markov Models (HMMs), the computation of the state likelihood is one of the most time consuming parts. In this paper, we propose novel multi-level Gaussian selection techniques to reduce the cost of state likelihood computation. These methods are based on original and efficient codebooks. The proposed algorithms are evaluated within the framework of a large vocabulary continuous speech recognition task.

Key words: Speech recognition, Gaussian selection, codebook

1 Introduction

The proliferation of mobile devices in daily life has created a great demand for efficient and simple interfaces. In particular, speech recognition being a key element of the conversational interface, there is a significant requirement for low-resource and accurate automatic speech recognition systems.
Recent mobile devices (GPS [1], GSM [2], PDA [3], ...) offer a large set of functionalities but their resources are still limited for accurate continuous speech recognition engines. Indeed, state-of-the-art continuous speech recognition systems use Hidden Markov Models (HMM) with many tens of thousands of Gaussian distributions to achieve improved recognition. The computation of

[1] GPS : Global Positioning System
[2] GSM : Global System for Mobile Communications
[3] PDA : Personal Digital Assistant

the emission probability of these Gaussian distributions is time consuming. As the performance and the speed of speech recognition systems are highly dependent on the number of HMM Gaussians, reducing the number of Gaussians without decreasing the system performance is of major interest (Suontausts et al. (1999); Chan et al. (2004); Sankar et al. (1999); Sagayama et al. (1995); Takahashi et al. (1995); Digalakis et al. (2000)).

According to previous studies (Bocchieri (1993); Mak et al. (2001); Jurgen et al. (1996); Gales et al. (1996); Chan et al. (2005)) only few Gaussians dominate the state likelihood computation. Hence, different techniques were developed to select them (Aiyer et al. (2000); Xiao et al. (2006); Kawahara et al. (2001); Filali et al. (2002); Sankar et al. (2002)). These techniques can be divided into two categories:

- *State based methods :* These methods aim to reduce the number of Gaussians per state. They are often applied to acoustic models with a high number of Gaussians per state such as semi-continuous or context independent models (Jurgen et al. (1995); Woszczyna (1998)).
- *Model based methods :* These methods are applied to models with a limited number of Gaussians per state such as triphones (Bocchieri (1993); Mak et al. (2001); Gales et al. (1999)). Their objective is to decrease the total number of Gaussians (belonging to all the states).

Gaussian selection is often performed in two steps : codebook construction (by Gaussian clustering) and Gaussian selection. In this paper, we propose several Gaussian selection techniques that reduce the cost of likelihood computation either at the state or model level. These techniques use a novel codebook construction process. The proposed methods are evaluated within the framework of a large vocabulary continuous speech recognition task and are compared to existing methods.

The document is organized as follows: section 2 describes the state based Gaussian clustering and selection. In particular, a weighted Kullback Leibler metric and several clustering criteria are introduced to improve the clustering. They are evaluated in the context of model shortening and used in a state-based Gaussian selection. Furthermore, a multi-level Gaussian selection algorithm is proposed. Section 3 describes our contributions to the model based approaches. Two mains propositions are detailed: contextual Gaussian selection and contextual sub-vector quantization. The conclusions and prospective work are described in section 4.

2 State-based clustering and selection

The state-based Gaussian selection is performed in two steps : classification and selection. During the first step, state Gaussians are grouped into clusters. Generally, they are organized into a tree structure based on their mean vector values (Ortmanns et al. (1997); Jurgen et al. (1996); Woszczyna (1998); Padmanablan et al. (1997)) and the Euclidian distance. The second step consists in selecting Gaussians to be used for the likelihood computation.
In this section, we propose a novel Gaussian clustering (ie., codebook construction) algorithm. This algorithm is based on a new metric and several empirical criteria. Then, we evaluate this clustering (and the new metric) within the framework of model shortening. Finally, we investigate the use of the new codebook in a state-based multi-level Gaussian selection.

2.1 Gaussian classification

For each Gaussian mixture, distributions are grouped into a binary tree structure and every cut in the tree defines a possible classification. To determine the optimal cut of the tree, two criteria are considered : data driven and dissimilarity based.

2.1.1 Clustering process

The bottom-up clustering algorithm is applied to each mixture of Gaussian distributions as follows :

1. Compute distances between each pair of distributions.
2. Merge the closest distributions : Let $g_1(n_1, \mu_1, \Sigma_1)$ and $g_2(n_2, \mu_2, \Sigma_2)$ be two Gaussians to which n_1 and n_2 frames have been assigned during the training. If g_1 and g_2 are merged into $g_3(n_3, \mu_3, \Sigma_3)$ then :

$$n_3 = n_1 + n_2 \tag{1}$$

$$\mu_3 = \frac{n_1}{n_1 + n_2}\mu_1 + \frac{n_2}{n_1 + n_2}\mu_2 \tag{2}$$

$$\Sigma_3 = \frac{n_1}{n_1 + n_2}\Sigma_1 + \frac{n_2}{n_1 + n_2}\Sigma_2 + \frac{n_1 n_2}{(n_1 + n_2)^2}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{3}$$

   g_3 replaces g_1 and g_2 in the set whose size is reduced by one (Mokbel (2001)).
3. If the number of Gaussians is greater than 1 go to the first step.

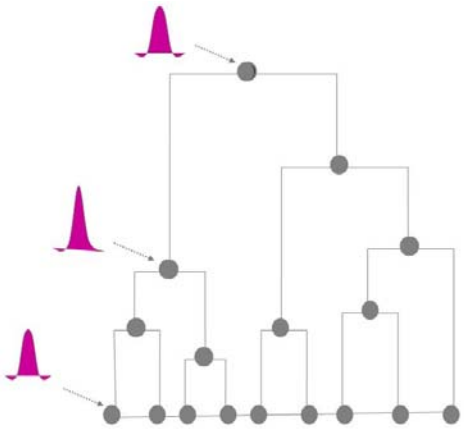

Fig. 1. Hierarchical clustering of Gaussian distributions.

2.1.2 Metrics

Two distances are used for Gaussian clustering: the existing likelihood loss-based distance (Mokbel (2001)) and the proposed weighted relative entropy based metric.

- *Loss likelihood based metric* : If g_1 and g_2 are merged into g_3 then the likelihood loss (PV) is the difference between the likelihoods of g_1 and g_2 and the likelihood of g_3 :

$$PV(g_1, g_2, g_3) = log \frac{\|\Sigma_3\|^{(n_1+n_2)/2}}{\|\Sigma_1\|^{n_1/2}\|\Sigma_2\|^{n_2/2}}$$

This metric is similar to the loss of entropy based distance used by Digalakis in (Digalakis (1996)). It was successfully used for model adaptation in (Mokbel (2001)).

- *The weighted symmetric Kullback-Leibler divergence (KLP)* : it is expressed as the distance between two probability density functions weighted by the amount of training data.

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

where d is the dimension of the parameter vectors and tr is the trace.
The information provided by the amount of training data is advantageous only if training and testing data have the same proportions.

2.1.3 Tree cutting

From the root of the tree to the leaves, cuts result in many different classifications (ie. codebooks). Three ways of cutting are proposed:

- *Fixed*: We consider a constant number of classes. So, the tree is traversed from the leaves until the number of nodes reaches the predefined number of classes.

- *Weight based*: The number of classes depends on the amount of training data in each class. So the tree is processed (from the root) and processing stops at the node whose children's weight is less than a predefined threshold.

- *Distance based*: The tree cutting is performed when the distance between two levels reaches a maximum value.

For weight and distance criteria, the number of Gaussians per state is variable. Hence, a mean value is computed.

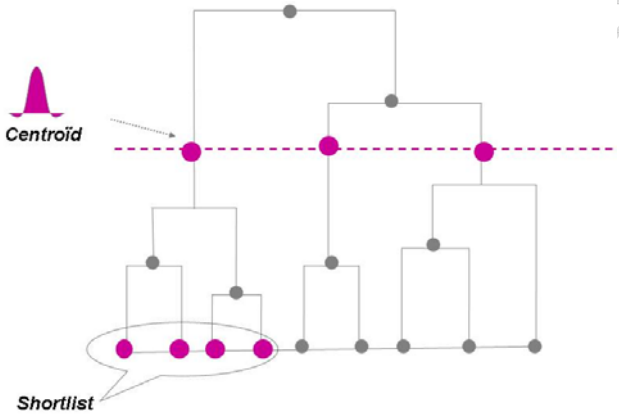

Fig. 2. Gaussian distributions classification

Codewords and *Centroids* refer to the nodes resulting from cutting the tree at a specified level. A *shortlist* is a set of tree leaves having a common *codeword*.

2.2 Model shortening

We propose to investigate the use of the previous clustering algorithm in model shortening. The proposed KLP metric (described in section 2.1.2) is also evaluated in the same context and compared to the loss likelihood distance (used in Digalakis (1996)).

For the experiments we use parameter vectors with 12 MFCC coefficients, en-

ergy, and their first and second derivatives. The acoustic models are context independent. They are trained with 82 hours of the Ester train database (Galliano et al. (2005)). The dictionary contains 118000 words and the language model is formed by 4 millions of bigrams and 4 millions of trigrams. An hour of Broadcast News extracted from the Ester test data set is used for testing.

In order to compare the different systems, a reference system with 32, 64, 80, 128, 180, 220 and 256 Gaussians per state is produced.
Considering the models with 256 Gaussians per state, the clustering algorithm is applied to each Gaussian mixture. Depending on the experiments, either the likelihood loss or the weighted cross entropy based metric is used. The previously explained tree cutting criteria (fixed, weight based or distance based) are also evaluated.

2.2.1 Fixed classes

The same number of classes is used by the reference system (REF) and by the loss likelihood (PV) and weighted Kullback-Leibler (KLP) based systems. After clustering, the PV and KLP models obtained are trained. We found that two iterations are sufficient for a suitable parameter estimation.Then speech recognition is applied to the REF, PV and KLP systems using models with 32, 64, 80,128 and 180 Gaussians per state. Results within a confidence interval of 1% are as follows:

Table 1
Fixed classes : WER for REF, PV, and KLP systems

Number of Gaussians	REF (%)	PV (%)	KLP (%)
32	42.6	40.6	39.5
64	40.4	38.0	37.5
80	38.3	37.4	36.9
128	37.3	36.2	36.2
180	36.4	36.1	36.2
220	36.3	35.8	35.5
256	36.3	-	-
512	35.5	-	-

An analysis of the results given in 1 and Fig. 3 shows that :

- Both the PV and the KLP systems outperform the reference one.
- WER decreases by about approximately 3% compared to the reference system.

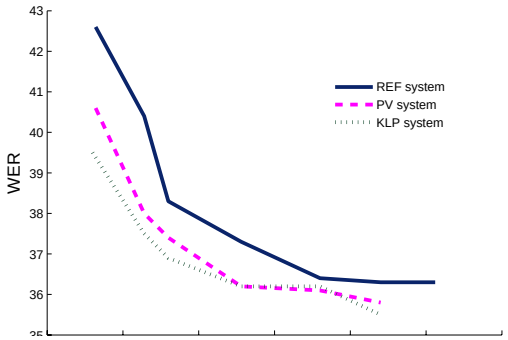

Fig. 3. WER vs number of Gaussians : Fixed classes for REF, PV and KLP systems

2.2.2 Weight based classes

By using this criterion, we ensure that each cluster has sufficient amount of training data for parameter estimation. As the number of Gaussians per state is variable (it depends on the acoustic variability of each state), a mean value is considered. Results are as reported in Table 2 and Fig. 4.

Table 2
Weight based classes : WER for PV and KLP systems

Metric	Number of Gaussians	WER (%)
KLP	28	40.0
	53	36.6
	150	35.9
	195	36.0
PV	53	39.5
	101	36.8
	156	36.5

We notice KLP outperforms both the PV and the reference system. Especially, with a mean of only 53 Gaussians per state, its performance is close to that of the REF system with 256 Gaussians per state. Besides, the WER decreases by about 4.8% compared to the REF system using the same number of Gaussians. 28 Gaussians per state on the KLP system perform better than 64 Gaussians in the reference system.

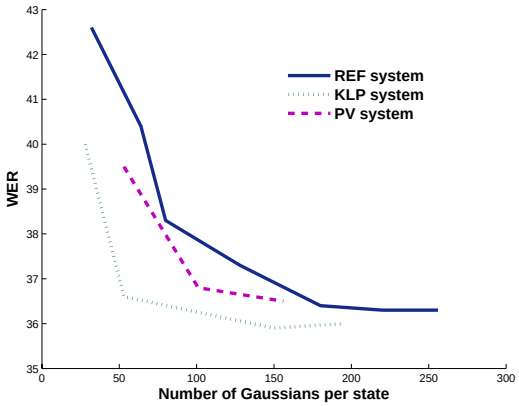

Fig. 4. WER vs number of Gaussians : Weight based classes for PV and KLP systems and fixed classes for REF system.

2.2.3 Distance based classes

This criterion prevents the clustering of too distant Gaussians. Gaussians are considered distant when their KLP distance is high or when their merging leads to a significant likelihood loss (if PV based metric is employed).

We consider several levels of the tree and cut when the distance between two levels reaches a maximum value. The obtained results are presented in Table 3.

Table 3
Distance based classes : WER for the PV and KLP systems

Metric	Number of Gaussians	WER (%)
KLP	30	40.7
	59	37.7
	101	36.1
	196	35.9
PV	44	39.4
	94	36.7
	204	35.8

Once again, we notice that the PV and KLP systems outperform the REF system, and that the KLP divergence based system has the lowest WER. Applying the KLP or PV clustering process, we obtain globally the same performance as the REF system using only about 40% of the total number of Gaussians. These results are encouraging but they are not better than the results of the previous experiments (53 Gaussians) in which only 20% of the initial Gaussians were used.

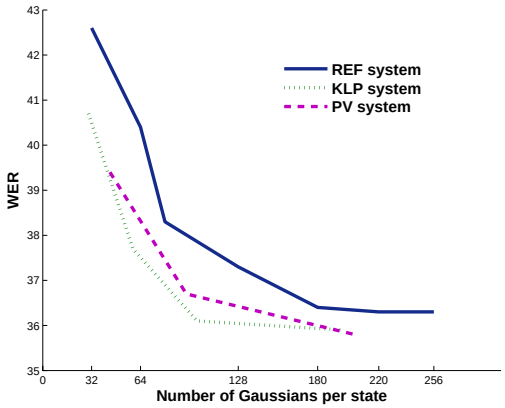

Fig. 5. WER vs number of Gaussians : Distance based classes for PV and KLP systems and fixed classes for REF system.

2.2.4 Weight versus distance

The goal of the previous experiments (sections 2.2.1, 2.2.2 and 2.2.3) is to compare the proposed KLP metric to the PV metric and to the REF system. Therefore we fixed the criterion and varied the distance. In this paragraph, the objective is to compare the distance criterion to the weight criterion. So we fixed the distance (KLP or PV) and plot the curves of the weight and the distance criteria.

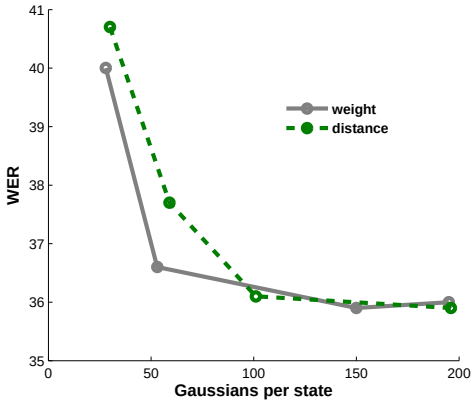

Fig. 6. KLP system : Weight and distance based tree cutting

An analysis of the results given in Fig. 6 and Fig. 7 show that for the KLP system, the weight criterion outperforms the distance criterion, particularly when the number of clusters is low. In the case of PV clustering, the situation is the contrary. These results can be interpreted as follows:

- When the KLP clustering metric is used, no particular attention is given to the amount of training data available for each cluster. Only similar Gaussians are merged, ensuring that at each level, clusters are as distant as pos-

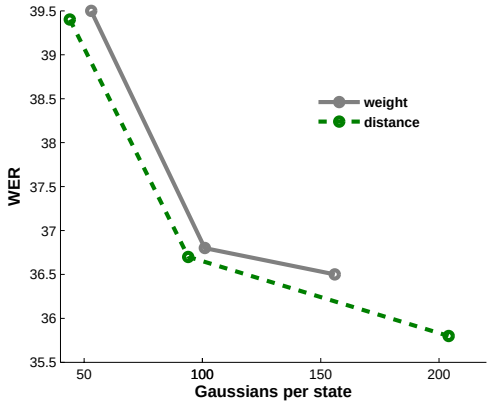

Fig. 7. *PV* system : Weight and distance based tree cutting

sible. therefore, at some levels, several clusters do not have enough training data, and cutting at these levels is of little value.

- In the case of *PV* based clustering, the loss of likelihood is minimal at each level. Therefore the resulting clusters are as representative as possible of the training data. Given that no information about cluster similarity is taken into account, many resembling clusters may be present at the same level. In this case the distance based cutting criterion is capable of removing redundant information

2.2.5 Conclusion

The clustering algorithm (presented in section 2.1) is investigated in the framework of model shortening.
Experiments show that the KLP distance performs better than the PV distance. When considering the tree cutting criteria, we notice that for the KLP system, the weight criterion outperforms the distance criterion and the for the PV system, the situation is the contrary.

2.3 State-based Gaussian selection

In the previous section (2.2) the clustering algorithm (described in section 2.1) is used to shorten the models by clustering the Gaussian distributions without decreasing the system performance. The second application of the clustering (and codebook construction) algorithm is in Gaussian selection. We investigate a likelihood-based, multi-level Gaussian selection.
The overall algorithm operates in two steps : in the first step, Gaussians are organized into a binary tree. Several cuts of the tree are performed. Each level of cut is characterized by its number of codewords.

In the second step (ie. selection) codeword likelihood is computed and sorted. Only the most likely codewords are considered when descending to the lowest cutting level. When the leaves of the tree are reached, the corresponding Gaussian distributions are sorted by weight and the best of them contribute to the likelihood computation.

2.3.1 The selection algorithm

Selection is applied during the decoding process. For each node of the decoding graph, the goal is to detect Gaussians that dominate the likelihood computation. It operates as follows :

(1) For the current pruning level, codeword likelihoods are computed. Then they are sorted and the most likely are kept before moving down to the lower level of cut.

(2) When reaching the last level of cut, 2 sets of Gaussian distributions may be selected for the likelihood computation :
   a) leaves whose ancestors have all been retained.
   b) leaves selected in a) with large weight values.

The following example (Fig. 8) illustrates an application of this algorithm to a mixture of 24 Gaussian distributions. In this case, two levels of cut are considered : level 1 and level 2.

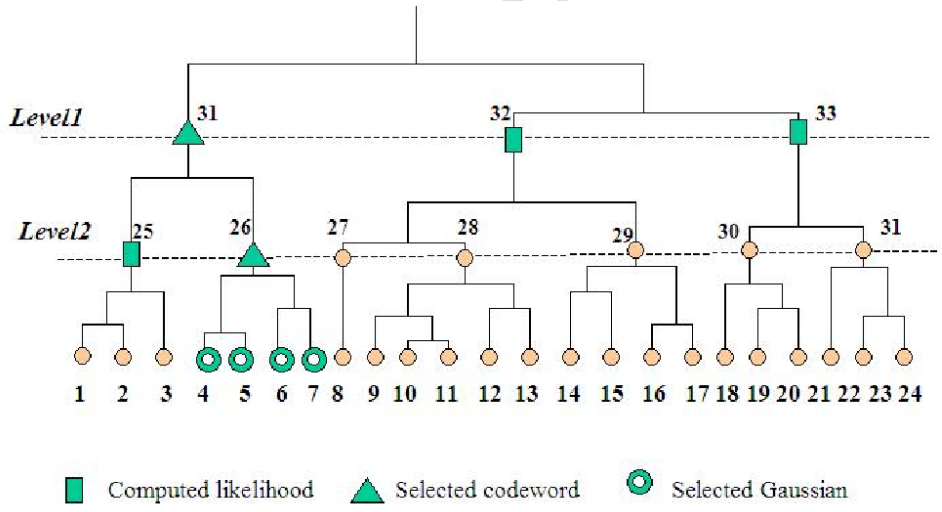

Fig. 8. Bi-level Gaussian selection example

First, likelihoods of the codewords 31, 32 and 33 are computed and sorted. As the codeword 31 is the most likely, it is selected. We then move to the next level of cut (level 2) and compute the likelihood of the corresponding nodes that are 25 and 26. If codeword 26 is more likely, the corresponding

leaves which are the Gaussians4,5, 6 and 7 are selected. Finally we may decide to compute the likelihood with all Gaussians or to keep only those with the highest weight values.
We thus computed a total of nine likelihoods which is less time consuming than 24.

2.3.2 One-level based selection

The reference system of the previous experiments (section 2.2) is considered. The acoustic models contains 512 Gaussian per state. For each state, the 512 Gaussian distributions are organized into a tree structure, then the tree is cut in a specific level. We experimented cutting the tree at the levels 40 and 120 which correspond respectively to 40 and 120 codewords. Performance is measured in terms of Word Error Rate (WER) and the percentage of likelihood computation C which is defined as :

$$C = \frac{computed \;\; likelihoods}{all \;\; likelihoods} \tag{4}$$

For the reference system : $WER = 35.5\%$ and $C = 100\%$

2.3.2.1 Shortlist scores : We vary the number of selected codewords (ie. centroids ; see Fig 2) and use the corresponding leaves (ie. shortlists) of the tree for the likelihood computation. The number of selected codewords, the corresponding number of Gaussians (X-axis) and WER (Y-axis) are reported in Fig. 9. As the number of the selected Gaussians is variable, a mean value is considered. The fraction C is also computed and depicted in Fig. 10.

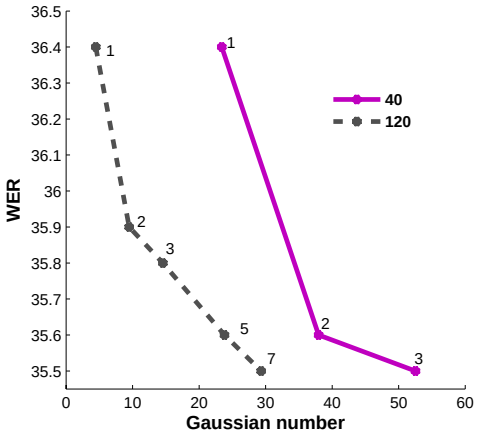

Fig. 9. Computing the likelihood using the best shortlists. The number in this figure correspond to the number of codewords.

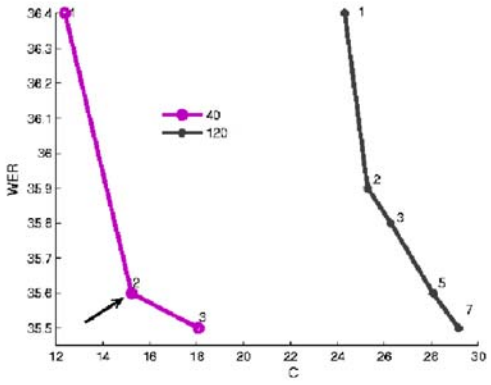

Fig. 10. Computing the likelihood using the best shortlists. The arrow shows the best tradeoff between C and the WER.

An analysis of the results given in Fig. 9 shows that for the same WER, the 120 codeword system makes use of fewer Gaussians than the 40 codewords one. In particular, with only 23 Gaussians per state (Fig. 9) it gives exactly the same results as the reference system.The same experiments (Fig. 10) show that the value of C is lower for the 40 codeword system. This is because this fraction takes into account the codebook size. The best tradeoff between C and the WER is obtained by the selection of 2 codewords (as shown by the arrow in Fig. 10). This corresponds to the value pairs (C,WER)=(15.1%,35.6%). In this case the WER increases by only 0.1% and the likelihood computation cost is reduced by a factor of seven. The acoustic matching duration of the reference system is 3 times the CPU time. This duration is decreased to 0.43 the CPU time.

2.3.2.2 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 Gaussian distributions with highest weights are retained. After varying the number of selected Gaussians, the results given in Fig. 11 are obtained.

The best tradeoff between C and WER is (12.4%,35,6%). These results are better than those of the previous experiments. Indeed, for the same value of WER (35.6%) the value of C is reduced. In this case, the likelihood computational cost is decreased by a factor of eight.

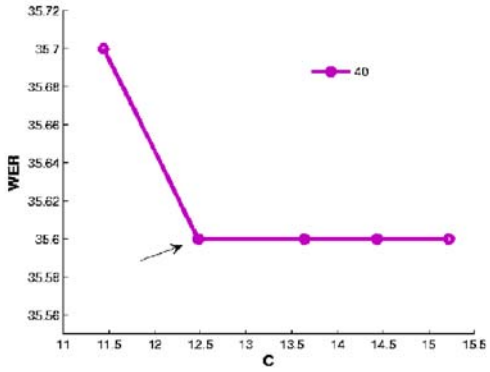

Fig. 11. Computing likelihood using the highest weight Gaussians. The arrow shows the best tradeoff between C and the WER.

2.3.3 Bi-level selection

Now the clustering tree is cut simultaneously at two levels (bi-level) of cut.Two experiments are considered :

- using the levels of cut 40 and 60 which correspond to 40 and 60 codewords.
- using the levels of cut 40 and 120 which correspond to 40 and 120 codewords.

2.3.3.1 Shortlist scores : In order to further improve the results of the experiments presented in section 2.3.2.1, all densities of level 40 are computed and the two best codewords are selected. Then we move 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 their codewords are used for the likelihood computation.

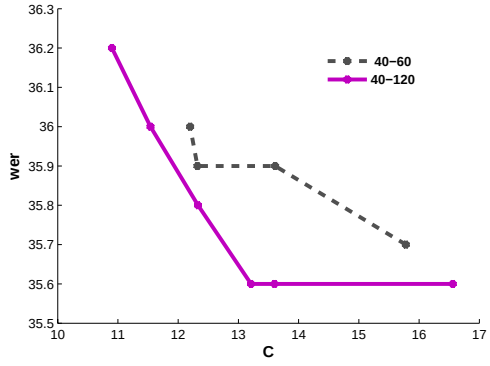

Fig. 12. Computing likelihood using two levels of cut and the best shortlists.

The 40-120 system gives better results than 40-60. This is foreseeable because the level 120 is lower than the level 60 therefore the classification is more precise. The best tradeoff between C and WER corresponds to the pair of values (C,WER) = (13,2%,35.6%). This result is better than the one presented

in section 2.3.2.1 where the tree was cut at a single level but poor than the result using weight values (section 2.3.2.2).

2.3.3.2 Data-based selection : We proceed in the same manner as in section 2.3.3.1. The best settings are considered : two levels of cut 40 and 120, and the best pair of values (C,WER)= (13,2%,35.6%). The optimization 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 given in Fig. 13.

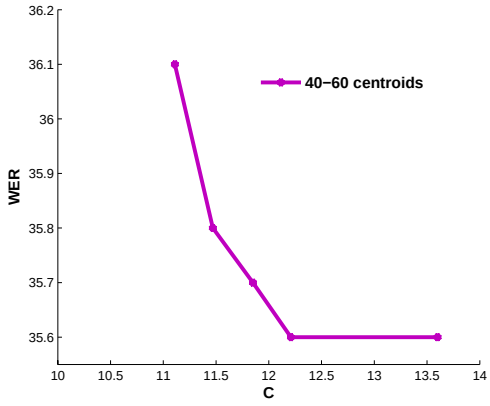

Fig. 13. Computing likelihood using the hightest weighted Gaussians.

Here, (C,WER)= (35,6%, 12.2%) is the best tradeoff between C and WER. As the width of the confidence interval is approximately 0.8%, other tradeoffs are also satisfactory. For example, in the case of pair values where (C,WER)= (35,8%, 11.5%) which denotes a decrease in the likelihood computation cost by a factor of nine with a negligible loss of accuracy (+0.3%).

2.3.4 Synthesis

Method	C (%)	$+WER(\%)$
one level based selection : shortlist scores	15.1	0.1
two levels based selection : shortlist scores	13.2	0.1
one level based selection : Data-based selection	12.4	0.1
two levels based selection : Data-based selection	12.2	0.1

Table 4
State based Gaussian selection results

We notice that :

a- Two levels selection performs better than one-level selection. So increasing the number of levels is advantageous.

b- The selection of Gaussians with high weight values (data-based selection) improves performance.
c- The combination of (a) and (b) gives the best results (best tradeoff between WER and C).

Since both the C values and the WER are very close, there is relatively little diffrence between any of the approaches.

3 Model based clustering and selection

To reduce the likelihood cost in HMM based ASR, two main approaches are generally used : Gaussian selection (Pellom et al. (2001); Lee (1997); Bocchieri (1993); Ortmanns et al. (1997); Herman et al. (1998))and sub-vector quantization. In both cases, classification is performed by clustering/merging Gaussian distributions (Bocchieri (1993); Ortmanns et al. (1997); Gales et al. (1999); Padmanablan et al. (1999)). So the contextual information is lost and some distributions will be assigned to codewords of different contexts.
Therefore we propose a context-based classification method. The idea is to use Gaussian distributions of context independent models as codewords. Then the clustering algorithm (previously described in section 2.1) is applied as a further improvement to the codebook. Indeed, this process provides a compact and more efficient codebook (see results in section 2.2).
Each codebook is tested within the framework of Gaussian selection and sub-vector quantization. The performance of the proposed Gaussian selection and sub-vector quantization methods are compared to two exiting methods which are evaluated in the same conditions.

3.1 Contextual Gaussian selection

Initially, Bocchieri (Bocchieri (1993)) proposed a Gaussian selection technique by vector quantization. He generates a vector quantized codebook and attributes a shortlist to each codebook entry. During decoding, the frame is assigned to the nearest codeword/shortlist. Gaussians belonging to this shortlist contribute to that frame likelihood computation. Many extensions of this work have been proposed in the literature (Gales et al. (1999); Olsen (2000); Leppänen et al. (2006)). They were focused on Gaussian codebook assignments. Here we are rather interested in improving codebook construction. The performance of the proposed methods are compared to the "classic Gaussian selection" method (proposed by Bocchieri in Bocchieri (1993)) and evaluated in the same conditions.

3.1.1 Gaussian distributions mapping

Large vocabulary continuous speech recognition systems need a significant number of Gaussians to model the different contexts. So Context dependent (CD) models are often used and the corresponding systems are generally slow. Small vocabulary systems make use of Context Independent models (CI) because the acoustic variability is limited. Hence, they have the advantage of being fast.
The CD and CI models may have been trained on the same data, though the have quite different capabilities for capturing it. Therefore, we investigate the use of CI Gaussian distributions as a codebook in a CD model based large vocabulary ASR system.

The proposed contextual Gaussian selection method consists in:

(1) Computing all the distances between the CI and the CD Gaussian distributions.
(2) Assigning each CD Gaussian distribution to the nearest CI.

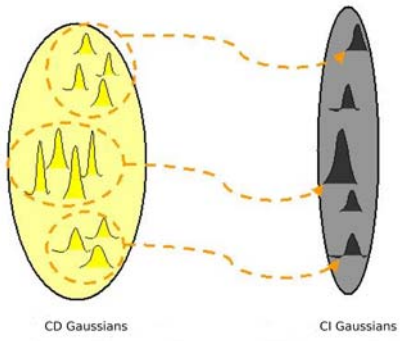

Fig. 14. Assignment of the CD Gaussian distributions to the CI Gaussian distributions.

In this case the codebook contains the CI Gaussian distributions and the CD distributions assigned to the same codeword (CI Gaussian distribution) form a shortlist.

3.1.2 Hierarchical mapping

The state based clustering algorithm (described in section 2.1) is applied to the CI distributions in order to reduce their length and improve their representation.

Subsequently, mapping table between the CD distributions and the new codebook (CI distributions after the clustering process) is created.

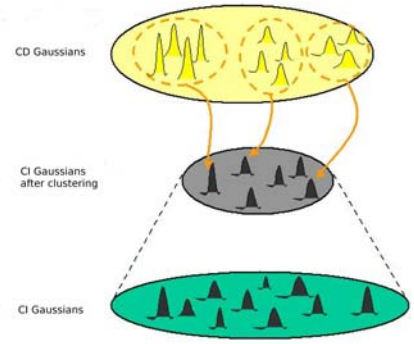

Fig. 15. Mapping between the new CI et CD Gaussian distributions. The new CI Gaussian distributions are obtained by clustering the initial CI Gaussian distributions.

3.1.3 Contextual selection

We developed a large vocabulary speech recognition system based on the Sphinx training and test tools. The acoustic models are cross-word context dependent with 6108 tied states and 32 Gaussians per state. The parameters of these models are estimated on the Ester training database. Tests are conducted using an hour of Broadcast News extracted from the Ester test set. For this reference system all the Gaussian distributions are used for the likelihood computation so $C = 100\%$. $WER = 28.7\%$.

Context independent models with 32 Gaussians per state and 3 states each were developed to perform contextual Gaussian selection. They have a total of 3456 Gaussian distributions (36*3*32).

After mapping the CD Gaussian distributions to the CI Gaussian distributions the classical Gaussian selection procedure (proposed by Bocchieri (1993) and described in section 3.1) was applied to the new codebook (formed by CI Gaussian distributions). This procedure is called Contextual Gaussian selection. The classical Gaussian selection method was also performed on a codebook obtained by Gaussian clustering (the same codebook as in Bocchieri (1993)) for comparative reasons. This codebook contains also 3456 Gaussian distributions. Fig. 16 reports the WER and C corresponding to varying the number of selected Gaussians for classic and contextual Gaussian selection.

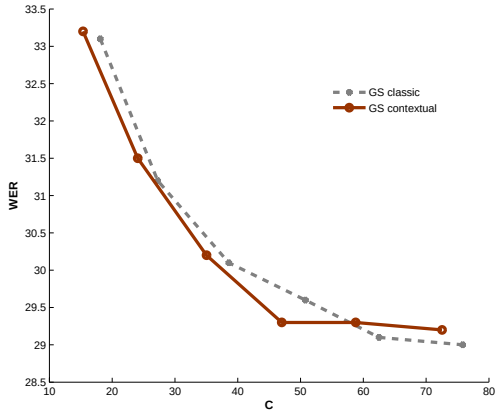

Fig. 16. WER vs C for classic and contextual Gaussian selection

We can see that contextual performs better than classic Gaussian selection. The best tradeoff between WER and C corresponds to the pair of values (29.3%,47.02%) and an absolute loss of accuracy of 0.6%.

3.1.4 Hierarchical selection

Hierarchical clustering (described in section 2.1) is applied to CI models with 64 and 128 Gaussians per state to be reduced to 32 Gaussians per state. Subsequently, classic Gaussian selection is performed using the new codebook.

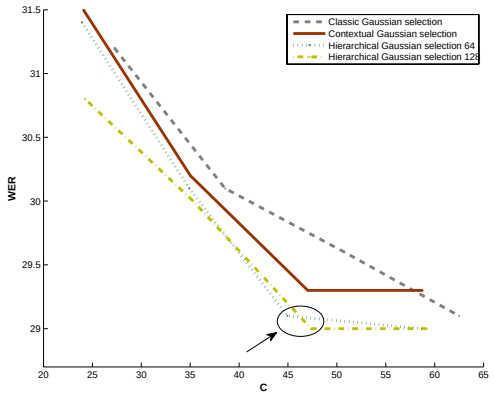

Fig. 17. WER vs C for classic, contextual and hierarchical Gaussian selection. The arrow shows the best tradeoff between C and WER.

From the results in Fig. 17, we notice that the WER of the models initially with 64 and 128 Gaussians per state are the lowest. Therefore, Gaussian clustering is also advantageous in sub-vector quantization. Typically, with 47% of computed likelihood the loss of accuracy is less than 0.3%.

3.1.5 Synthesis

From the following table we notice that :

- Contextual Gaussian selection performs better than classic Gaussian selection.
- Hierarchical Gaussian selection gives the best results.

Method	C (%)	Δ (WER) (%)
Classic Gaussian selection	50.7	+0.9
Contextual Gaussian selection	47.0	+0.6
Hierarchical Gaussian selection	47.5	+0.3

Table 5
Model based Gaussian selection results

The initial speech recognition systems runs in 4 * CPU time. When contextual Gaussian selection is performed, the recognition duration is reduced to 3 * CPU time.
In addition, there is some benefit in terms of slightly imroved recognition.

3.2 Contextual sub-vector quantization

Recently, sub-vector quantization based methods were proposed as an alternative to the Gaussian selection approach. They have been successfully applied to reduce acoustic model complexity without significant loss of accuracy (Mak et al. (2001); Tsakalidis et al. (1999); Mosur et al. (1997)). Several methods for codebook construction have been investigated. They are generally based on clustering techniques. For example Mak(Mak et al. (2001)) performs per stream Gaussian clustering by means of Battacharya distance. A speech group at Carnegie Mellon University (CMU) employs the k-means algorithm to cluster sub-vectors (means and variances) into a preset number of codebooks (Mosur et al. (1997)), ..
As contextual information is lost by clustering, we are interested (in this subsection) in contextual sub-vector quantization. We subsequently investigate the improvement of the codebook by hierarchical clustering. Results are compared to those obtained by the method described in (Mosur et al. (1997)) which was evaluated in the same conditions.

3.2.1 Stream-based mapping

The contextual sub-vector quantization method is performed in two steps:

(1) the mean and variance vectors of each CI and CD distribution are divided into streams (i.e. subsets of dimensions).
(2) for each stream, the symmetric Kullback-Leibler distances between the CD and CI distributions are computed.
(3) each CD distribution is assigned to the closest CI distribution.

By the end of this process, we obtain a per stream mapping table between the CD and CI distributions.

In Mosur et al. (1997), the codebook is created by simultaneously clustering the streams of mean and variance vectors. The K-means algorithm is employed for the clustering.

The parameter vectors are composed of (12MFCC + energy) and their first and second derivatives. Three subdivisions of the parameters vector are considered:

- Only one stream of dimension 39.
- Three streams: (12MFCC+energy) + Δ(12MFCC+energy)+ Δ Δ(12MFCC+energy).
- Four streams : (energy,Δenergy,Δ Δenergy) + (12 MFCC) + (12 Δ(MFCC)) + (12 Δ Δ (MFCC)).

In the following, the performance of the contextual sub-vector quantization method are compared to those of the initial system which are (WER, C) = (28.7%, 100%) and to those of the existing sub-vector quantization method (described in Mosur et al. (1997)).

3.2.2 Contextual Sub-Vector Quantization

For the contextual sub-vector quantization (CSVQ), 36 CI models with 32 Gaussians per state (a total of 108 states) are used for the codebook. The mapping is performed by state (CSVQ-s), by phone (CSVQ-p) or using all the CI distributions (CSVQ-a).

- CSVQ-s : the Gaussian distributions of each monophone state constitute a codebook for the corresponding triphone states.

- CSVQ-p : the Gaussian distributions of a monophone constitute a codebook for the corresponding triphones.

- CSVQ-a : the Gaussians distributions of all the monophones constitute a codebook for all the triphones.

During the decoding process, the likelihood is computed with the CI distributions and the corresponding CD distribution weights.

Fig. 18 reports the WER according to the computation fraction C for the methods SVQ (the method described in Mosur et al. (1997)), $CSVQ-s$, $CSVQ-p$ and $CSVQ-a$ and for the three sizes of stream (1, 3 and 4).

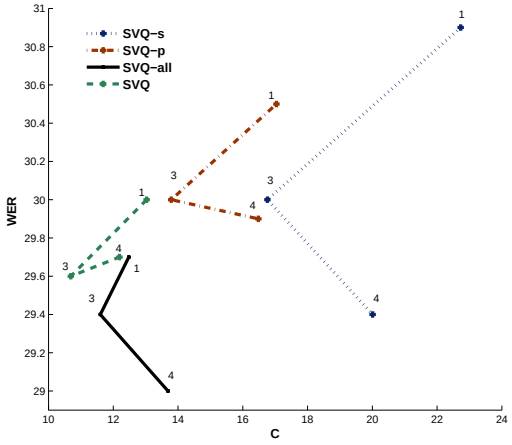

Fig. 18. The performance of the SVQ and CSVQ methods for the three streams

Several observations can be made :

- Multi-stream based methods perform better than one-stream based methods. Indeed, the quantification distorsion (ie. error due to assigning (sub)vectors to codewords) is less important.
- The increase in WER generated by the SVQ, $CSVQ-s$ and $CSVQ-p$ methods exceeds 1%. From our point of view, this is due to the loss of information about context in the SVQ method. For the $CSVQ-s$ and $CSVQ-p$ approaches, we can say that the distributions of the CI states and phones are unable to represent all the corresponding CD distributions.

- The best results are obtained by means of the CSVQ-a method. We point out that this method makes use of a codebook formed by all of the CI distributions. The optimal configuration corresponds to the pair of values $(WER, C) = (29.0\%, 13.6\%)$, i.e. the likelihood computation fraction is reduced to 13.6% with a small increase in the WER (+0.3% absolute). The initial speech recognition systems runs in 4 * CPU time. When CSVQ-a method is performed, this duration is reduced to 2.8 * CPU time.

3.2.3 Hierarchical Sub-Vector Quantization

To improve the results of the $CSVQ-a$ method, we applied the clustering algorithm to the CI Gaussian distributions. The length of the $CSVQ-a$ codebook is 3456 (ie. 32 Gaussians * 108 states). It was reduced to 540 (ie.

5 Gaussians * 108 states). $CSVQ - h$ refers to the new system with a total of 540 Gaussians. xtures. To compare the results of these experiments to the previous ones, we use the same stream definitions.

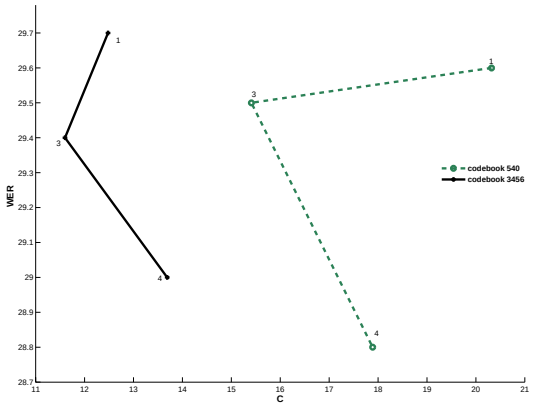

Fig. 19. The SVQ-a and CSVQ-h results for the three streams

From the results in Fig. 19, we can deduce that :

- CSVQ-h method outperforms SVQ-a.
- Using only one stream is of little interest (the WER increase exceeds 0.9% absolute).
- The CSVQ-h method produces an interesting point (WER,C)=(28.8%,17.89%) which compares well with the initial system. In addition, the WER is inside the confidence interval and thus we can conclude that about 17% of densities are computed with no loss of accuracy.

4 Conclusion

In order to reduce the computation cost in low-resource and large application mobile devices several Gaussian selection and sub-vector quantization techniques were investigated. These methods are based on original and efficient codebooks and operate either at the state or at the model level.
Two kinds of codebook have been proposed. The first is formed by context independent Gaussian distributions. The second results from clustering context independent Gaussian distributions. The weighted and symmetric Kullback-Leibler distance is used for the clustering.
The evaluation of the codebooks and the selection methods has been conducted within the framework of a large vocabulary continuous speech recognition task. Experiments shows that by using the first codebook, the WER decreases by 3% absolute. For the same WER the likelihood computation cost is reduced to 17%.

As a perspective, we propose to combine these Gaussian selection and subvector quantization techniques for a further improvement of performance.

5 Acknowledgments

The authors would like to thank Peter Weyer-brown, Kevin McTait and Hemant Misra for their corrections of English orthography and syntax.

References

Aiyer, A., Gales, MJF., & Picheny, MA., Rapid Likelihood Calculation of Subspace Clustered Gaussian Components, International Conference on Acoustics, Speech, and Signal Processing, 2000, 1519-1522

Bocchieri, E., Vector Quantization for the Efficient Computation of Continuous Density Likelihoods, International Conference on Acoustics Speech and Signal Processing, 1993, 692-695.

Chan, A., Sherwani, J., Mosur, R., & Rudnicky, A., Four Layer Categorization Scheme of Fast GMM Computation Techniques in Large Vocabulary Continuous Speech Recognition Systems, International Conference on Spoken Language Processing, Jesu Island - Korea, 2004,

Chan, A., Ravishankar, M., & Rudnicky, A., On Improvements to CI based GMM Selection, European Conference on Speech Communication and Technology, Lisbon, September 2005, 565-568

Digalakis, V., Monaco, P., & Murveit, H., Genones : Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers, IEEE Transactions on Speech and audio Processing, 1996, vol. 4, 281-289

Digalakis, V., Tsakalidis, S., Harizakis, c., & Neumeyer, L., Efficient Speech Recognition using Subvector Quantization and Discrete-Mixture HMMs, Computer Speech and Language, 2000, 33-46

Filali, K., Li, X., & Bilmes, J., Data-driven Vector Clustering for Low Memory Footprint ASR, International Conference on Spoken Language Processing, 2002

Gales, MJF., McKnill, K., & Young, S., State based Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMMs, IEEE Transactions on Speech and Audio Processing, 1999, 470-473

Gales, MJF., McKnill, K., & Young, S., Use of Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMMs, International Conference on Spoken Language Processing, 1996, 470-473

Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, JF., & Gravier, G., The Ester Phase II Campaign for the Rich Transcription of French

Broadcast News, European Conference on Speech Communication and Technology, 2005

Herman, S. H., & Sukkar R. A., Joint MCE estimation of VQ and HMM parameters for Gaussian Mixture Selection, IEEE International Conference on Acoustics, Speech and Signal Processing, 1998, 485-488.

Jurgen, F., & Ivica, R., The Bucket Box Intersection (BBI) Algorithm for Fast Approximative Evaluation of Diagonal Mixture Gaussians, International Conference on Acoustics Speech and Signal Processing, 1996, 837-840

Jurgen, F., Ivica, R., & Tile, S., Speeding up the Score Computation of HMM Speech Recognizers with the Bucket Voronoi Intersection Algorithm, European Conference on Speech Communication and Technology, 1996, 1091-1094

Lee, A., Kawahara, T., Takeda, K., & Shikano, K., A New Phonetic Tied-Mixture Model for Efficient Decoding, International Conference on Acoustics Speech and Signal Processing, 2001, 1269-1272

Lee, A., Gaussian Mixture Selection using Context Independent HMM, IEEE International Conference on Acoustics Speech and Signal Processing, 1997

Leppänen, J., & Kiss, I., Gaussian Selection with Non-Overlapping Clusters for ASR in Embedded Devices, International Conference on Acoustics Speech and Signal Processing, 2006

Li, X., Malkin, J., & Bilmes, J., A High-speed, Low-Resource ASR Back-end based on Custom Arithmetic, IEEE Transactions on Audio, Speech and Language Processing, 2006

Mak, B., & Bocchieri, E. Subspace Distribution Clustering Hidden Markov Model, IEEE transactions on Speech and Audio Processing, 2001, 264-275

Mokbel, C., Online Adaptation of HMMs to Real Life Conditions: A Unified Framework, IEEE Transaction on Speech and Audio Processing, 2001, 342-357

Mosur, R. , M., Bisiani, R., & Thayer, E., Sub-vector Clustering to Improve Memory and Speed Performance of Acoustic Likelihood Computation, European Conference on Speech Communication and Technology, 1997, Rhodes, Greece

Olsen, J., Gaussian Selection using Multiple Quantisation Indexes, IEEE Nordic Processing symposium, 2000

Ortmanns, S., Ney, H., & Firslaff, T., Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition, European Conference on Speech Communication and Technology, 1997, Rhodes, Greece, 139-142.

Padmanablan, M., Jan, E., Bahl, L. & Picheny M., Decision-Tree based Feature Space Quantization for Fast Gaussian Computation, IEEE Workshop on Automatic Speech Recognition and Understanding, 1997, Santa Barbara, 325-330

Padmanablan, M., Bahl, L., & Nahamoo, D., Partitioning the Feature Space of a Classifier with Linear Hyperplanes, IEEE Transactions on Speech and Audio Processing, 1999, 282-288

Pellom, B., Sarikaya, R., & Hansen, J.H.L., Fast Likelihood Computation Techniques for Nearest-Neighbor Based Search for Continuous Speech Recognition, IEEE Signal Processing Letters, 2001, vol. 8, no. 8, 221-224

Sagayama, S., & Takahashi, S., On the Use of Scalar Quantization for Fast HMM Computation, International Conference on Acoustics, Speech, and Signal Processing, 1995, vol. 2, 213-216

Sankar, A., Ramana, V., Slolcke, & Weng, F., Improved Modeling and Efficiency for Automatic Transcription of Broadcast News, Speech Communication, 2002, vol. 37, 133-158

Sankar, A., & Ramana, V., Parameter Tying and Gaussian Clustering for Faster, Better, and Smaller Speech Recognition, European Conference on Speech, Communication and Technology, Grece, 1999

Suontausts, J., Hakkinen, J., & Viikki, O., Fast Decoding Techniques for Practical Realtime Speech Recognition Systems, Workshop on Automatic Speech Recognition and Understanding, 1999,

Takahashi, S., & Sagayama, S., Four-level Tied-structure for Efficient Representation of Acoustic Modeling, International Conference on Acoustics, Speech, and Signal Processing, 1995, vol. 1, 520-523

Tsakalidis, S., Digalakis, V., & Neumeyer, L., Efficient Speech Recognition using Subvector Quantization and Discrete-Mixture HMMs, IEEE International Conference on Acoustics, Speech and Signal Processing, 1999, 569-572

Woszczyna, M., Fast Speaker Independent Large Vocabulary Speech Recognition, PHD Thesis, Karlsruhe University, 1998,