GAUSSIAN DENSITY TREE STRUCTURE IN A MULTI-GAUSSIAN HMM-BASED SPEECH RECOGNITION SYSTEM

J. Simonin, L. Delphin-Poulat & G. Damnati

France Télécom - CNET - DIH/DIPS
Technopole Anticipa
2, Avenue Pierre Marzin, 22307 Lannion - France

ABSTRACT

This paper presents a Gaussian density tree structure usage which enables a computational cost reduction without a significant degradation of recognition performances, during a continuous speech recognition process.

The Gaussian tree structure is built from successive Gaussian density merging. Each node of the tree is associated with a Gaussian density, and the actual HMM densities are associated to the leaves. We propose then a criterion to decide whether a node belonging to a high level in the tree should be expanded or not. The expansion means that the likelihood is evaluated with Gaussian densities associated with a low level node if the likelihood computed at the high level is not precise enough.

This Gaussian tree structure is evaluated with a continuous speech recognition system on a telephone database. The expansion criterion allows a 75 to 85% computational cost reduction in terms of log-likelihood computations without any significant word error rate increase during the recognition process.

1. INTRODUCTION

The efficiency of a continuous speech recognition system depends on the trade-off between recognition performances and computational cost. In order to improve recognition performances, the number of acoustic parameters is often increased with multi-Gaussian output distributions in a classical Hidden Markov Model (HMM). A Gaussian density splitting procedure is used to increase the number of Gaussian densities for each distribution during the parameters training [1]. However, the computational cost increases with the number of acoustic parameters.

An efficient way to reduce computational cost is to decrease the total number of log-likelihood evaluations for each observation. A Gaussian selection method succeeding to a Gaussian density clustering may be applied [2].

An other way to reduce computational cost is the use of a Gaussian density tree structure [3]. A clustering algorithm was proposed in [3] to build a tree structure. Then, during the recognition process, the likelihood was only calculated for the N most likely densities at each level. In this paper, we propose a binary Gaussian tree building algorithm, and a tree search method allowing to calculate log-likelihood during the Viterbi decoding from only two levels in the tree.

Hence, the paper is organized as follows :

First, the study aims to use a Gaussian density merging technique to build a Gaussian density tree structure. Here, a bottom-up clustering strategy is applied to build a binary tree by successive Gaussian densities merging. Indeed, Gaussian components clustering of multi-Gaussian distributions is a classical way to tie density parameters [4].

Then, we propose a way to use this tree structure during the speech recognition process. Two levels are useful in the tree : the lower level corresponding to initial Gaussian densities and a higher level. We define a criterion to decide which node associated to a Gaussian density should be used for log-likelihood computation. The aim is to obtain equivalent speech recognition performances with a lower computational cost.

2. GAUSSIAN TREE STRUCTURE BUILDING

We choose a bottom-up strategy to build a binary Gaussian density tree structure.

2.1. Gaussian Tree Building Algorithm

Successive merging of Gaussian densities allow a binary tree to be built. Indeed, a parent node represents the result of the merging of two Gaussian densities representing the two child nodes into a single Gaussian density. Gaussian parameters associated with a parent node are defined by merging formulas in the next paragraph. Successive merging of Gaussian densities associated with parentless tree nodes are applied to build parent nodes until the tree root node.

The binary Gaussian tree structure building algorithm is then :

- The low level is defined by nodes associated with actual HMM Gaussian densities. This level is actually formed by the leaves of the tree.
- Until the tree root node, different levels are built with following principles :
  - merging concerns only parentless nodes,
  - a Gaussian density associated with a parent node created at a given level ρ can not be merged at the same level ρ with another density,

- the distance between two Gaussian densities associated with two child nodes must be lower than an upper bound, at each level, to allow a merging between these two densities.

This Gaussian density tree is the result of successive merging of HMM Gaussian densities.

2.2. Gaussian Density Merging

The HMM output distribution for a frame X[τ] at time τ related to a transition is given by:

$$B\,(X[\tau]) = \max_{1\le k\le NG} \{c_k.N(X[\tau];\, \mu_k,\, \Sigma_k)\}$$

where $N(.;\mu_k, \Sigma_k)$ is a Gaussian density with a mean vector μ_k, a diagonal covariance matrix Σ_k , and c_k, the Gaussian component weight. NG is the number of Gaussian components of the multi-Gaussian distribution B.

A clustering strategy is used to merge Gaussian densities. At each step of the building algorithm, the closeness of all pairs of Gaussian densities is evaluated in order to achieve the appropriate merging. $N(.;\ \mu_1,\ \Sigma_1)$ and $N(.;\ \mu_2,\ \Sigma_2)$ denote two Gaussian functions, with μ_1 and μ_2 mean vectors, Σ_1 and Σ_2 diagonal covariance matrices, to which n_1 and n_2 acoustical frames have been associated during the training corpus. The distance between these two functions is measured as the decrease in the likelihood of the corresponding training set observation after merging [5]. If d is the acoustic space dimension, the distance D is given by:

$$D = -n_1 \,.\, \sum_{i=1}^{d} \log(\sigma_{1i}) - n_2 \,.\, \sum_{i=1}^{d} \log(\sigma_{2i}) + (n_1 + n_2) \,.\sum_{i=1}^{d} \log(\sigma_i)$$

where $(\Sigma_1) = (\sigma^2_{1i})_{(1\le i\le d)\}}$, $(\Sigma_2) = (\sigma^2_{2i})_{(1\le i\le d)}$. $(\Sigma) = (\sigma^2_i)_{(1\le i\le d)}$ is the diagonal parameters vector of the covariance matrix resulting from the merging.

If these two Gaussian functions are merged, the resulting function has a number of frames equal to the sum of the number of frames associated to the functions that are merged. Its parameters μ_i and σ^2_i after a weight normalization, are estimated by:

$$n'_1 = n_1 \,/\, (n_1 + n_2)$$
$$n'_2 = n_2 \,/\, (n_1 + n_2)$$
$$\mu_i = n'_1 \,.\, \mu_{1i} + n'_2 \,.\, \mu_{2i}$$
$$\sigma^2_i = n'_1 \,.\, \sigma^2_{1i} + n'_2 \,.\, \sigma^2_{2i} + n'_1 \,.\, n'_2 \,.\, (\mu_{1i} - \mu_{2i})^2$$

2.3. Gaussian Tree Structure Description

This building algorithm produces a tree structure which can be described as follows.

A tree node at level ρ, TN(ρ, n, μ, Σ), is associated with a Gaussian density defined by classical Gaussian density parameters, i.e. weight n, mean vector μ and covariance matrix Σ.

In Figure 1, a Gaussian density tree structure is characterized by a low level and a high level. Moreover, this example shows that for each level, all nodes at a level (here the low) are not necessarily merged because of the third principle of the binary tree building algorithm.

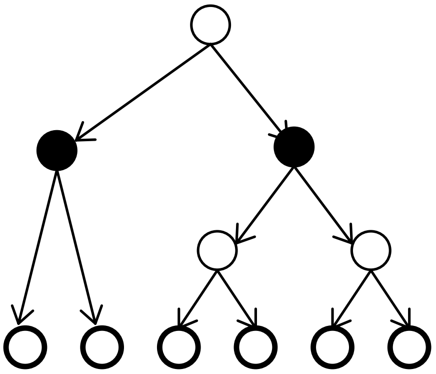

Figure 1 : Gaussian density tree structure with a high level (bullet) and low level (bold ring), and where each parent node is the result of the merging of two child nodes.

After the Gaussian density tree building, only high level and low level nodes are used during the recognition process. Indeed, log-likelihood evaluated with a Gaussian density associated to high level node may be enough for descendant log-likelihood estimation, if this density is unlikely. The Gaussian tree structure with its two specific level allows a decrease of the log-likelihood computations.

3. GAUSSIAN TREE STRUCTURE USAGE DURING THE VITERBI DECODING

The problem to solve during the Viterbi decoding is to obtain good recognition performances with a Gaussian density tree structure.

3.1. Gaussian Tree Usage Algorithm

During the recognition process, a beam search strategy is applied [6] . That means that only log-likelihood for Gaussian density associated with active model transitions are estimated.

Recall that the aim is the efficient use of the multi-Gaussian tree structure to decrease the number of log-likelihood evaluations considering emission distributions and observations. Therefore, we choose two particular levels in the tree. First, a high level is defined in the Gaussian tree structure, corresponding to an a priori chosen number of nodes, i.e. a percentage of the total number of initial Gaussian densities. The low level corresponds to leaves of the tree.

During the Viterbi decoding, for each frame, a maximum is evaluated among the different high level nodes log-likelihoods. Then, the nodes of this high level for which the log-likelihood is close to this maximum are expanded. This expansion to their corresponding low level nodes allows a more precise evaluation of the log-likelihood.

The algorithm may be described as follows for each frame $X[\tau]$:

- Log-likelihood evaluation of Gaussian densities associated with a high level node.
- Log-likelihood threshold (LLT) estimation for high level nodes which determine high level nodes to be expanded.
- *If* a low level node is a descendant of a high level node which has to be expanded, *then* log-likelihood for the Gaussian density associated to the low level descendant must be evaluated, *else* the previously estimated high level node log-likelihood is used for the Gaussian density associated to the low level nodes.

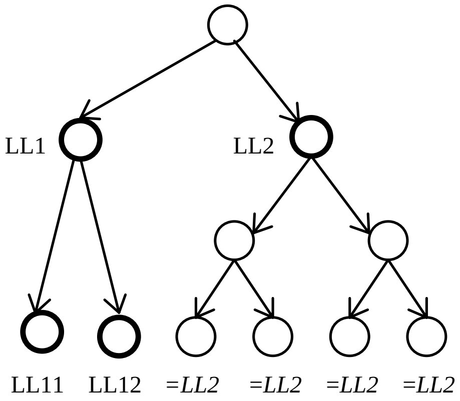

Figure 2 : Example of a Gaussian density structure usage during the Viterbi decoding with log-likelihood estimation (LLN) associated to each node.

Figure 2 means that only 4 Gaussian densities, associated to bold nodes, are used during log-likelihood estimation in the recognition process for this example. Indeed, only the left node of the high level is expanded. Four log-likelihood estimations with the tree structure are required during the Viterbi decoding (2 at the high level and 2 at the low level) instead of 6 log-likelihood estimation.

3.2. Criterion for Node Expanding

The criterion allows to expand a high level tree node. It consists in determining a threshold LLT over the likelihood estimation which applies to each high level node as described above. We propose a criterion derived from the Minimum Description Length [7]. This criterion is used to select nodes which have to be expanded during the Viterbi decoding. The length is for a recognition process the likelihood estimated for an observation and a HMM with multi-Gaussian distribution.

This threshold is evaluated as follows :

Let X denotes the observation sequence, λ_h the parameters of the model corresponding to the high level, M_l the number of low level nodes and M_h the number of high level nodes. We determine a maximum MLL over the log-likelihood for the high level nodes allowing to compute the threshold LLT.

$$\mathrm{MLL}\ (\{X\}, \lambda_h) = \max_{1 \le k \le M_h} (\log(P(\{X\}|TN(\rho, n_k, \mu_k, \Sigma_k))))$$

$$\mathrm{LLT}\ (\{X\}, \lambda_h) = \mathrm{MLL}\ (\{X\}, \lambda_h) + \frac{\alpha}{2} \log(M_h/M_l)$$

α is the number of free parameters subject to estimation. Gaussian density parameters are mean vector and diagonal covariance matrix parameters. $\log(P(\{X\}|TN(\rho, n_k, \mu_k, \Sigma_k)))$ represents the log-likelihood evaluated for the k^{th} node at the ρ^{th} tree level associated with the Gaussian density $N(.;\mu_k, \Sigma_k)$.

If the log-likelihood estimated at a high level node is greater than LLT, this high level node must be expanded. That means that its low level child nodes need an accurate log-likelihood estimation with initial Gaussian density parameters.

The proposed criterion allows to expand a number of nodes which increases as the total number of high level nodes decreases. This is coherent with the fact that, the less nodes are used, the less precise are the log-likelihood evaluations, the less restrictive should the criterion be. Another property deduced from the MDL criterion is that the delta between the log-likelihood maximum evaluated at the high level and LLT is not function of the log-likelihood maximum but of the total number of high and low level nodes.

4. EXPERIMENTS

This tree structure reduces the number of log-likelihood evaluations to be computed. The approach is evaluated using a continuous speech recognition system over a human-machine dialogue speaker-independent telephone database [8].

4.1. Training and Test Databases

The training corpus for the task-independent part consists of about 700 short sentences recorded by hundred of speakers calling from different regions of France. This telephone database contains almost all the French diphones. Moreover, a task-dependent part is made of 5451 sentences containing 26111 words recorded by speakers dialoging with AGS dialog system. The AGS dialog system vocabulary contains 876 words.

An evaluation of the system is achieved on a task of voice services directory inquiry about weather forecasts and employment. A telephone database with 724 sentences containing 3584 words is obtained. The speech recognition system used is obviously speaker-independent.

4.2. Gaussian Tree Structure and Criterion Evaluation

The evaluation of the Gaussian density tree structure consists in speech recognition tests with an a priori high level. This level has a specific number of nodes in the tree. The node expansion with the previously described criterion is applied at many levels of the tree to check the criterion efficiency.

The three tables below present the evaluation results in terms of word error rate and computational cost reduction. This computational cost reduction is evaluated by means of the number of log-likelihood computations using a tree structure (Gaussian density associated with a high level node included) compared to the number of log-likelihood computations using only actual Gaussian densities, i.e. associated with the low level node.

The first experiment (Table 1) involves a total number of 3210 Gaussian densities in the HMM. This case corresponds to a single Gaussian component for each Gaussian distribution.

HLN	*3210*	1402	817	440	220	55
WER	*24.6%*	25.6%	25.5%	25.5%	25.9%	25.8%
CCR		31.3%	59.0%	73.9%	75.3%	42.9%

Table 1 : word error rate (WER) and computational cost reduction (CCR) with a specific number of high level nodes (HLN) (1 component per distribution).

A first noticeable point is that the word error rate never decreases significantly (95% confident interval). A second point is a maximum computational cost reduction of 75.3% for the number of log-likelihood estimations.

Other evaluations (Tables 2 & 3) are achieved with a total number of Gaussian densities multiplied by 2 and 4. These cases correspond to a number of Gaussian components by distribution equal to respectively 2 and 4.

HLN	*6243*	2952	1852	1060	574	73
WER	*22.6%*	23.4%	23.6%	22.8%	22.0%	22.9%
CCR		20.1%	50.0%	70.0%	79.5%	46.3%

Table 2 : word error rate (WER) and computational cost reduction (CCR) with a specific number of high level nodes (HLN) (2 components per distribution).

HLN	*11915*	4056	2436	1377	751	195
WER	*21.6%*	21.3%	22.1%	21.8%	21.0%	21.4%
CCR		43.8%	66.0%	79.1%	84.0%	66.7%

Table 3 : word error rate (WER) and computational cost reduction (CCR) with a specific number of high level nodes (HLN) (4 components per distribution)..

In terms of computational cost reduction, the tree structure enables an important reduction of about 80%. Moreover, the property of no significant word error rate increase is confirmed (95% confident interval).

Also, we observe a slight word error rate decrease which we can compare with previous multi-Gaussian densities merging study [1]. Indeed, Gaussian density merging with multi-Gaussian distributions (2, 4 or 8 components by distribution) allows word error rate improvement if the total number of Gaussian densities is divided by 2.

Finally, for the three evaluations, there is an optimum for the number of high level nodes which is between 6% and 9% of the total number of Gaussian densities.

5. CONCLUSION

Experiments show a 75 to 85% computational cost reduction in terms of log-likelihood evaluation without any significant increase of word error rate. This evaluation with different total number of Gaussian density shows the efficiency of the proposed criterion.

Moreover, we observe that reducing the number of Gaussian densities when the number of Gaussian components per distribution is more than one, may improve the word error rate.

6. RÉFÉRENCES

1. J. Simonin, S. Bodin, D. Jouvet & K. Bartkova, “Parameter tying for flexible speech recognition”, *Proc. ICSLP*, Vol. 2 : 1089-1092, Philadelphia, PA, USA, 1996.

2. K.M. Knill, M.J.F. Gales & S.J. Young, “Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs”, *Proc. ICSLP*, Vol. 1 : 470-473, Philadelphia, PA, USA, 1996.

3. T. Watanabe, K. Shinoda, K. Takagi, K.I. Iso, “High speed speech recognition using tree-structured probability density function”, *Proc. ICASSP*, Vol. 1 : 556-559, Detroit, Michigan, USA, 1995.

4. M.Y. Hwang, X. Huang “Shared-Distribution Hidden Markov Models for Speech Recognition”, *IEEE Trans. Speech and Audio Processing* 1: 414-420, 1993.

5. D. Jouvet, L. Mauuary, J. Monné “Automatic Adjustments of the Structure of Markov Models for Speech Recognition Applications”, *Proc. of EuroSpeech*, Vol.2: 923-927, Genova, Italy, 1991.

6. V. Steinbiss, B. Tran & H. Ney, “Improvements in beam search”, *Proc. ICSLP*, Vol. 4 : 2143-2146, Yokohama, Japan, 1994.

7. J. Rissanen, “A universal prior for integers and estimation by minimum description length”, *Ann. Statistics*, Vol. 11, N°2 : 416-431, 1983.

8. M.D. Sadek, A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, J. Simonin, “Effective Human-Computer Cooperative Spoken Dialogue : the AGS Demonstrator”, *Proc. ISSD* : 169-172, Philadelphia, PA, USA, 1996.