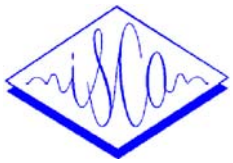
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

COMPACT ACOUSTIC MODEL FOR EMBEDDED IMPLEMENTATION

Junho Park *Hanseok Ko*

ISPL, Department of Electronics Engineering, Korea University, Seoul, Korea
jhpark@ispl.korea.ac.kr hsko@korea.ac.kr

10.21437/Interspeech.2004-273

ABSTRACT

An acoustic model for an embedded speech recognition system must exhibit two desirable features; ability to minimize performance degradation in recognition while solving the memory problem under limited system resources. To cope with the challenges, we introduce the state-clustered tied-mixture (SCTM) HMM as an acoustic model optimization. The proposed SCTM modeling shows a significant improvement in recognition performance as well as a solution to sparse training data problem. Moreover, the state weight quantizing method achieves a drastic reduction in model size. In this paper, we describe the acoustic model optimization procedure for embedded speech recognition system and corresponding performance evaluation results.

1. INTRODUCTION

Speech recognition technology as a prominent human-machine interface scheme has been used in many useful applications. However, speech recognition applications must overcome the difficulties posed by the working environment and system constraints. In realizing an embedded system application, our main focus is on achieving portability and performance reliability (e.g. word error rate) under constrained system resources.

To achieve these two objectives, many HMM-based acoustic modeling methods for embedded system have been investigated. Context independent (CI) model alone does not well represent the acoustic unit in all contexts though its size is relatively small. As a result, CI models do not achieve high recognition performance for large vocabulary recognition tasks. On the other hand, context dependent (CD) models provide a more detailed description of the acoustic units undergoing analysis. However, the gain in the detailed acoustic models requires an adequate amount of training data and memory size. Because the memory storage in embedded system is usually limited, we can no longer port the CD model into the system. This problem can be, in part, resolved by employing either tied-mixture (TM) continuous parameter, phonetic tied-mixture (PTM), or subspace distribution clustering (SDC) modeling [1][2][3].

Generally, tied-mixture context dependent HMM modeling method needs solutions for the training data sparse problem, similar to that of the continuous HMM. One of the most popular methods is "parameter smoothing." The parameter smoothing is to generate reliable parameter values of each CD model using existing CI model and less reliable (or before smoothing) CD model. One weak point of these smoothing methods is algorithm complexity for finding the smoothened CD model parameter values. However, the state-clustered tied-mixture (SCTM) provides itself as a solution to finding reliable CD model parameter values with various state clustering algorithms. PTM method is an alternative tied-mixture modeling focused to represent more detailed Gaussian codebook. PTM method, however, has many more free parameters to estimate than SCTM. As a consequence, the parameter convergence rate is relatively slow and more training data set for this model is needed. The SDC model was investigated aiming at achieving much more compact acoustic models based on tying the parameters of the subspace distribution unit of continuous density Gaussian mixtures.

In the next section we first describe the proposed state clustered tied-mixture (SCTM) model. We then prescribe in section 3 the optimizing method of SCTM for embedded speech recognition system. In Section 4, we present the representative experiments that show how effective the proposed SCTM model and our optimizing method are for embedded speech recognition system. Finally, concluding remarks are presented in Section 5.

2. STATE CLUSTERED TIED-MIXTURE HMM

The semi-continuous HMM (SCHMM) has state output probabilities compromised between discrete and continuous distribution HMM's. That is, the codebook of the SCHMM, which has Gaussian code-words, is constructed by a vector quantization (VQ) process, and its state output probability is the weighted sum of these code-words. Eq. (1) shows the output probability of state S for input vector $\mathbf{x}_t$.

$$b_S(\mathbf{x}_t) = \sum_{l=1}^{L} b_S(l) N(\mathbf{x}_t; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \qquad (1)$$

where L is the size of the codebook, $b_S(l)$ is the weight for each codebook index l, and $N(\bullet)$ denotes Gaussian distribution.

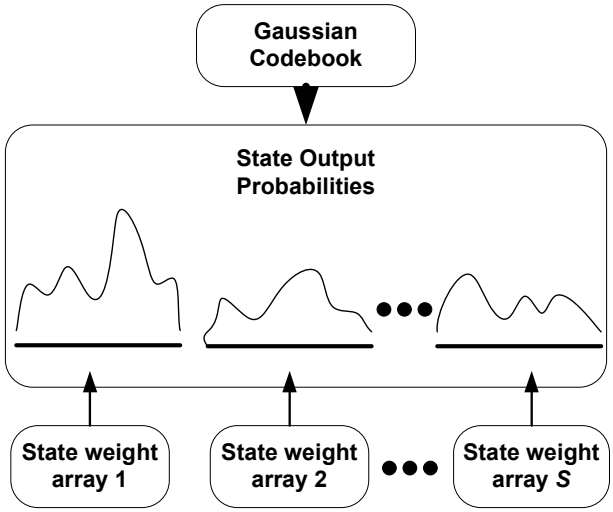


Figure 1: Concept of tied-mixture modeling method

Tied-mixture HMM modeling method: The tied-mixture modeling method is used to make one or more codebooks for tying Gaussians from a continuous density Gaussian mixture. The continuous distribution HMM (CDHMM) has a Gaussian mixture and corresponding weights for each state. Fig. 1 shows the concept of tied-mixture (TM) modeling.

Construction of state clustered tied-mixture HMM: Co-articulation effects for several contexts can be represented by context dependent HMM (e.g. triphone). The context dependent HMM must share their parameters such as the state distributions for reliable parameter estimation. In Fig. 1, some state weight arrays can be clustered in the same class. As a result,, the SCTM model shares some state output distributions while its state shares common Gaussians in the codebook. This SCTM model has properties of both the state-clustered and the tied-mixture models. Additionally, in the state clustering process, the SCTM model may generate decision-trees for unseen contexts. Fig. 2 shows the structure of the SCTM model.

In the SCTM HMM, the tied Gaussians are trained using the data frames from many states but the state weights are estimated with data only if the state shared. In other words, using the same amount of data, one can improve the state modeling as well as estimate the tied Gaussians [4].

3. OPTIMIZING SCTM MODEL

The SCTM based acoustic model for an embedded speech recognition system has various optimizing targets. The objectives are; to reduce the decoding complexity, to increase decoding speed, and to compact the acoustic

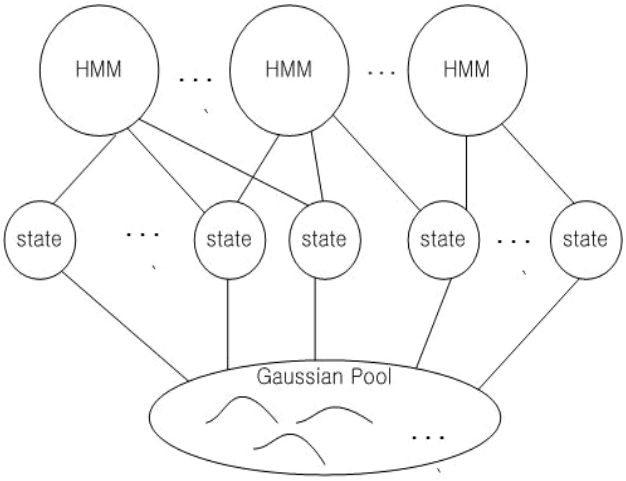


Figure 2: Structure of state clustered tied-mixture HMM

model for portability to an embedded system. In this section, we introduce our methods for optimizing the SCTM model.

Lower boundary of state weight array: In the model training procedure, we define the lower boundary with $2x10^{-5}$. The lower boundary prevents the state weights from falling into zero. Once state weights fall to zero in any iteration, they cannot return to meaningful values. These zero state weight becomes a significant factor in increasing the word error rate (WER) of the system. The lower bounded state weights have values in the range -10.82 ~ 0 on a log scale. In other words, the variance of state weights is relatively small to that of any other parameter values. Small variance of state weight enables quantization into a finite level.

State weights quantization: In the SCTM model, the largest part of a model size is the set of state weight arrays since the number of codeword is no less than that of Gaussian mixture of CDHMM and all states have the same count of weights as that of codeword in the codebook. The data type of state weight arrays is floating point. Consequently, a 4-byte memory is needed per a weight value. However, if we quantize the weights into 256 levels, each weight needs only 1-byte memory. Thus, the memory reduction in the SCTM model is more pronounced than that of PTM. As a consequence, the weight quantized SCTM model size can be reduced to the size of reduced PTM model size, though the original (before quantizing) model size of PTM is smaller than the SCTM model size. Furthermore, if an embedded system can accommodate some memory caused by state weights quantization, we can easily append special states for extra vocabulary [6].

State level Gaussian selection: The aim of Gaussian selection (GS) is to reduce the computational load by selecting the subset of Gaussian component likelihoods that should be computed, given a particular input vector [5]. The motivation behind GS is as follows: If an input vector is an outlier with respect to Gaussian component distribution, then the likelihood of that Gaussian component producing the input vector is very small. This results in the likelihood that the input frame per Gaussian component within a state will have a large dynamic range, with one or two Gaussian components tending to dominate the state likelihood for a particular input vector. Hence, the state likelihood can be computed solely from these Gaussian components, without a noticeable loss in accuracy. However, the GS in TM based models is less effective than the GS in CDHMM based models since the total number of Gaussians is very small and all states are sharing those Gaussians. Moreover, in the TM based model, the probability of Gaussians in the codebook are pre-computed only once per frame input. On the other hand, the time for log-likelihood additions to each state is very large since the number of additions is equal to the number of code-words or state weights. Thus, the time spent to compute the Gaussian probabilities for inputs is relatively small in terms of total computation time. As a consequence, we use a state level Gaussian selection algorithm to maximize the effectiveness of GS for the SCTM model. The state level Gaussian selection algorithm uses a sub-set of pre-computed Gaussians for getting each output probability. The Gaussian selection criterion we use is likelihood difference of each Gaussian to the maximum likelihood valued Gaussian. If there are too many Gaussians satisfying the criterion, we just use the appropriate number of Gaussians for state output probability. Eq. (2) represents the state level Gaussian selection algorithm.

$$b_S^{GS}(\mathbf{x}_t) = \sum_{\substack{N(\mathbf{x}_t;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \\ \in criterion}} b_S(i) N(\mathbf{x}_t;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \qquad (2)$$

where *criterion* means the subset of Gaussian likelihood satisfying the above selection criterion and any other parameters are the same as Eq. (1).
Gaussian components of satisfying *criterion* associated with each state are defined using the following selection routine

$$\begin{aligned} &N(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \in criterion \text{ iff} \\ &N(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \in \underset{\mathrm{i}}{\operatorname{argmin}}(n)\, E\{D(b_S(\mathbf{x}_t), b_S^{GS}(\mathbf{x}_t))\} \end{aligned} \qquad (3)$$

where $D(b_S(\mathbf{x}_t), b_S^{GS}(\mathbf{x}_t))$ represents the difference between fully computed and Gaussian selected state probability, while $\operatorname{argmin}(n)$ resents the minimum n Gaussian components selected.
Using this state level Gaussian selection algorithm can reduce much of the computational load used to compute the state output probability since the number of used Gaussians for adding log-likelihoods is reduced.

4. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed acoustic model implementation method for an embedded system using the SCTM model and corresponding optimizing methods.

Comparison of the SCTM and the CDHMM models: In this experiment, the SCTM model is a 2-stream, 256-codebook sized per stream, and tree-based state clustered tied-mixture HMM, State weights of the SCTM model are not yet quantized. The CDHMM for comparison is a 2-stream, 16-mixture per state, tree-based state clustered continuous distributed HMM. The total number of states of SCTM and CDHMM is identical to each other. The feature we used is 26th order MFCC's constructed with 12th order MFCC's, log energy, and their delta values.
In this test, we used a set of 452 phonetically balanced Korean words set both as the training model and test DB in order to simplifying the process. For speaker independent tasks, we precluded the test speaker's utterances from those in training. Table 1 shows both recognition performances and model sizes of each model. In the context independent modeling, the TM model showed poor results since both WER and model size came to being less efficient than CDHMM's. However, in the context dependent modeling for improving WER, the SCTM model showed drastic reduction in model size although its WER increased only slightly.

Table 1: WER and model size of SCTMs and CDHMMs

		Context Independent	Context Dependent
WER	CDHMM	5.94 %	0.96 %
	SCTM	6.57 %	1.15 %
Model Size	CDHMM	219 kB	1,354 kB
	SCTM	303 kB	811 kB

Performance of the state level GS algorithm: In this test, we evaluated the performance of the state level GS algorithm. First, we tested this algorithm in a clean test environment. We then added a test for car noisy environment with a noise reduction pre-processing. The training DB we used was 64,000 utterance set for car navigation system and the test DB was 4000 utterance set collected under real driving test. The SCTM model structure was same as the model of above test. Table 2

presents the WER and decoding time comparisons the state level GS algorithm applied system with baseline system. Here, baseline system means the system using all of Gaussians for computing the state output probabilities. 'Clean' and 'Noisy' reflects the environment in which the engine of the car is off and is running at the speed of 60~80 km/h, respectively. We applied spectral subtraction and RATZ algorithm to the noisy test as a noise reduction pre-processing. This pre-processing step was not included in computing the decoding time. The WER of the state level GS system in both environments is slightly increased when compared to the baseline. However, average decoding time is significantly reduced. The average decoding time of the noisy test is larger than that of the clean test since the number of Gaussians satisfying the criterion defined in Section 3 in noisy test is on average larger than that in the clean test. In short, noise reduces discriminative properties of Gaussians.

Table 2: Performances of state level GS applying system

System		Baseline	State level GS
WER	Clean	0.89%	1.23%
	Noisy	6.93%	7.53%
Decoding Time	Clean	T	0.57T
	Noisy	T	0.63T

State weights quantization of the SCTM model: In this test, we verified the performance of state weight quantizing and its properties. All of the test setting are identical with the above test. Table 3 shows the variations of WER and model size corresponding to the state weight quantizing levels. When we quantized the state weights to 256 levels (1-byte), WER as well as model size reduction were preserved. However, in the case of 128 levels (0.5-byte), WER started to increase.

Table 3: Performances of state weights quantizing

System	Baseline (state level GS)	256 levels (1-byte)	128 levels (0.5-byte)
WER	1.23%	1.23%	1.69%
Model size	1,733 kB	544 kB	340 kB

5. CONCLUSTIONS

In this paper, we implemented an acoustic model based on a state clustered tied-mixture HMM structure, targeting for embedded speech recognition systems. Moreover, we investigated optimization techniques, such as lower bounded state weights training, state level Gaussian selection, and state weights quantizing. These optimization schemes made our model more effective and portable. In particular, the state level Gaussian selection algorithm was shown to be suitable for the proposed SCTM model and the envisioned speech recognition system. The algorithm was shown to reduce the average decoding time. Namely, the computational load for achieving state output probabilities can be decreased and a real-time embedded system was implemented.

Experimental results show that our SCTM model can be applied to any small-size embedded speech recognition system. In state weight quantizing test, we expect an improved WER performance under small quantization if we can find the levels using maximum likelihood level estimation from all of the weights.

6. REFERENCES

[1] Young, S. J., "*The general use of tying in phoneme-based HMM speech recognizers,*" ICASSP*1992, pp. 569- 572.*

[2] Lee, A., et al, "*A new phonetic tied-mixture model for efficient decoding,*" ICASSP*2000, pp. 1269-1272.*

[3] Bocchieri, E. and Mak, B. K-W., "*Subspace distribution clustering hidden Markov model,"* IEEE Trans on speech and Audio Processing*, vol. 9, March 2001, pp. 264-276.*

[4] Duchateau, J., et al, "*Fast and accurate acoustic modeling with semi-continuous HMMs,*" Speech Communication, v*ol 24, 1998, pp. 5-17.*

[5] Gales, M., et al, "*State-based Gaussian selection in large vocabulary continuous speech recognition using HMM's,"* IEEE Trans on speech and Audio Processing*, vol. 7, March 1999, pp.152-161.*

[6] Kim, K.H., et al, "*Usage of exclusive HMMs for Korean-Digit in the embedded ASR system using tied-mixture method," Submitted to Proc.* ICSLP'04*, 2004.*