Reduced Semi-continuous Models for Large Vocabulary Continuous Speech Recognition in Dutch*

K. Demuynck, J. Duchateau and D. Van Compernolle †

K. U. Leuven - E.S.A.T., Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium
E-mail: Kris.Demuynck@esat.kuleuven.ac.be, Jacques.Duchateau@esat.kuleuven.ac.be

ABSTRACT

Semi-continuous Density HMM's have - due to the decoupling between the set of gaussians and the other HMM-parameters - more possibilities than Continuous Density HMM's to match the number of parameters in the model to the available train data. The computational load of the SC-HMM's however is huge compared to the load of their continuous counterparts, because of the large mixture weighting vector and because of the fact that for each frame all gaussians have to be evaluated. This paper describes the different steps taken to reduce the computational load of the SC-HMM's, resulting in faster and better models.

1. Introduction

Our current research in the field of speech recognition focuses on the development of a HMM-based large vocabulary speaker-independent continuous speech recognition system for Dutch. This paper deals with the acoustic modeling.

We opted for Semi-continuous Density Hidden Markov Models [2] as they have a better decoupling between the number of gaussians and the number of states than Continuous Density Hidden Markov Models. This is a nice feature as more data is required to reliably estimate a gaussian than there is data needed to estimate the mixture weights and transition probabilities. The additional flexibility of the SC-HMM can thus be used to match the number of parameters in the HMM to the size of the available train data for both the gaussians and the other HMM-parameters independently.

The evaluation of the observation density functions for the Semi-continuous models is very time consuming however, due to the large mixture weighting vector and due to the fact that for each frame all gaussians have to be evaluated. A solution to both problems is proposed in this paper.

The remainder of this paper is organised as follows: in section 2, the different steps taken to reduce the size of the mixture weighting vector are described. Next, in section 3, a novel algorithm to select the most promising gaussians - without evaluating them - is proposed. In the last sections, the resulting models are compared with CD-HMM's on both an Isolated Word Recognition (IWR) task and a Continuous Speech Recognition (CSR) task, and some conclusions are given.

* This research was supported by IWT Research Contract 940044, entitled Nerex.

† Currently with Lernout & Hauspie Speech Products.

2. Reduced SC-HMM's

The output probability density functions in our continuous HMM's are modeled as a weighted sum of gaussian density functions. In formula the output probability of state j for frame $\overline{X}$ is given by

$$P_j(\overline{X}) = \sum_{i=1}^{N} \lambda_{ij} \times \mathcal{N}_i(\overline{X})$$

with λ_{ij} the weight for gaussian i in state j and $\mathcal{N}_i(\overline{X})$ the probability of gaussian i. This section does not consider the calculation of the gaussian probabilities, i.e. we suppose here that they are given.

The remainder of the calculations is proportional to N, the number of gaussians used in one state. For CD-HMM's this number is small and these calculations are neglectable w.r.t. the calculations of the gaussian probabilities. SC-HMM's on the other hand use all gaussians in all states. Therefore N is very large and the cost to calculate the weighted sum now becomes by far the most important one.

The solution to this problem is to make a weighted sum of only a subset of all gaussians. The selection of this subset can be done in two ways: data driven or model driven.
In the **data driven** method, only the best gaussians are selected for a given data frame, the rest of the gaussian probabilities $\mathcal{N}_i(\overline{X})$ is ignored in the weighted sum. This selection does not depend on the state. It is further explained in section 2.1.
The **model driven** method only selects the best gaussians for a given state, the rest of the weights λ_{ij} is set to zero. This selection does not depend on the given data frame. It is done before the recognition starts and results in what we call a reduced SC-HMM (see section 2.2).

2.1. Top-N calculations

The first method to simplify the calculation of the weighted sum works as follows. For an input data frame, first calculate the probabilities of all gaussians. Then select the N gaussians with the highest probabilities (the Top-N gaussians). Use only this subset of gaussians to calculate the weighted sums for all states. The extra cost of this method is the sorting of the gaussian probabilities.

2.2. Reduced SC-HMM's

This method modifies the SC-HMM. For each state j the N gaussians with the highest weights λ_{ij} are selected. The result is a reduced SC-HMM. During recognition, the fixed subset of gaussians

for each state can be used, thus reducing the calculations for the weighted sums without any extra cost.

The reduction of a full SC-HMM to a reduced SC-HMM with the final number of gaussians in a state is not done in one training step, but with a sequential procedure. The reasoning is as follows. Omitting a number of gaussians in a state implies that the remaining gaussians have to model the frames that contributed to the omitted gaussians. So the weights for the remaining gaussians (and their order) can change due to the reduction. Therefore successive, smaller reduction steps are needed to guarantee well-trained weights and consequently a good selection.

The complete reduction algorithm as used for our experiments is described below.

1. Start with an initial full SC-HMM. The set of gaussians is copied from a CD-HMM. The value of the initial weights is of little importance.
2. Do a first training pass with the full SC-HMM using the Top-N system for the weighted sums in the output probability calculations (32 is a good value for N).
3. Reduce the SC-HMM with one large step to 128 gaussians per state and do a second training pass. This large reduction step can be taken because for a given state, a lot of gaussians are not useful at all. Moreover training steps with a large number of gaussians per state are very slow.
4. Further reduce the number of gaussians per state with steps of a factor two until the final value is reached (64 or 32 in our case). Add one training pass after each reduction.
5. Finally do two additional training passes on this SC-HMM without further reduction.

Note that it is possible that by reducing a SC-HMM, a certain gaussian is not used in any of the states. In this (rare) case, the gaussian is removed from the gaussian set.

3. The evaluation of the gaussians

The computational cost for the reduced SC-HMM's described above lies almost entirely with the evaluation of the gaussians. Therefore we searched for an algorithm to select the most promising gaussians in far less time than needed to actually evaluate them.

3.1. Proposed solutions in the literature

In [1] an algorithm is proposed to find the best density function out of a given set of density functions. The algorithm has some drawbacks however: it only guarantees to find the best gaussian and the distance metric has to be the same for all density functions. This means that all gaussians must have the same variance.

Another approach is described in [5]. The gaussians are clustered in 16 groups and for each group an envelope gaussian is calculated. For each frame, only those gaussians belonging to the N best scoring envelope gaussians are evaluated. For N equals 5, i.e. an effective decrease of the computational load with a factor 2.7, a neglectable drop in recognition performance was noticed.

Instead of selecting the most probable gaussians, one can also simplify the likelihood calculation. In [4], most floating point operations are replaced by integer operations by using scalar-quantisation on the input vector. A second reduction was achieved by truncating the calculation whenever a gaussian was found to be unlikely in one direction.

3.2. Fast Removal of Gaussians (FRG)

The main problem with the clustering algorithm [5], is the high dimensionality of the input vector. A good selection algorithm should be able to decide at each point whether a gaussian is likely enough or not. Along each axis in the data space there are several changes in the likelihood ordering of the gaussians, so each axis should be divided into several regions. This however leads to a tremendous number of divisions for the high dimensional parameter vectors used in speech recognition.

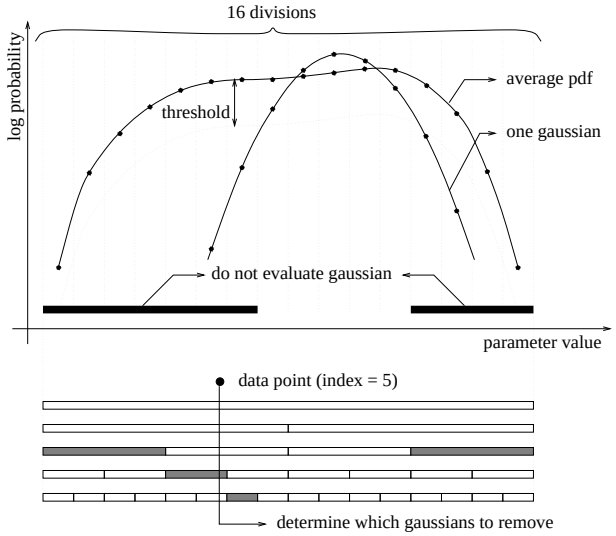

Figure 1: A one-dimensional solution to the problem of detecting unlikely gaussians

Therefore we opted for a one-dimensional approach. Each axis is divided linearly in a large number of cells. A gaussian is marked as non-likely for given axis and cell if its probability in the centre of the cell is less than a specified fraction (e.g. 1/20) of the average probability over all gaussians in the HMM (see figure 1). During recognition, a gaussian will be marked as non-likely for a given frame, if it is marked as non-likely for at least one of the axes. The same fraction (or threshold when log probabilities are considered) is used for all axes. The number of gaussians that will be removed can be controlled by varying this threshold.

To calculate the probability for a gaussian along a given axis, we use the marginal distributions. As N-dimensional gaussians with a diagonal covariance matrix are the product of N one-dimensional gaussians, the marginal density functions are easily derived:

$$\mathcal{N}(x_i, \mu_i, \sigma_i) \;=\; \frac{1}{\sigma_i\sqrt{2\pi}} \frac{(x_i - \mu_i)^2}{\sigma_i^2}$$

When combining the information over the different axes, the resulting regions described by the gaussian and the likely/unlikely marks start to differ (see figure 2). Particularly the tails of the distributions are modeled differently. As long as the recognition system does not

have to rely on these tail distributions (no severe mismatch between train and test conditions), the selection will be fairly good.

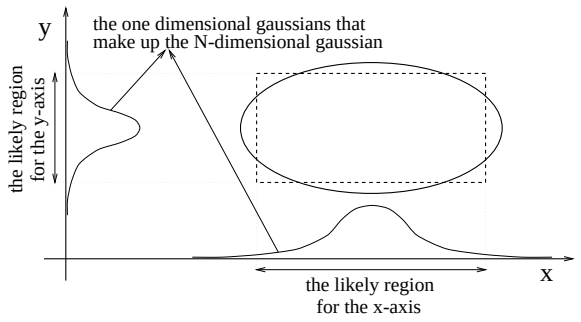

Figure 2: The region described by a gaussian (solid line) and the region described by the combined likely/unlikely-marks (dashed line)

3.3. Storing the information

A good representation of the likely/unlikely-marks should give us a fast access to the list of all gaussians that are **NOT** to be evaluated. In this way, there is almost no additional cost for all gaussians that have to be evaluated. For unlikely gaussians, the cost of evaluating the gaussians is replaced by the cost of setting a flag one or more times to zero. The trivial solution - i.e. listing all unlikely gaussians for each division and axis - requires too much memory however.

Therefore we describe the regions as the exclusive union of simple binary regions (see figure 1). To each binary region a list is attached containing the gaussians that are unlikely over the complete range of the binary region. This organisation reduces the memory requirements strongly, while the lists of unlikely-gaussians are still easily accessible. If we assume that the unlikely-regions are trivial, i.e. they have at most two borders, then we have the following upper bound on the number of gaussian indexes that have to be stored for each axis:

$$2 \times N_{gauss} \times \log_2(N_{div}),$$

with N_{div} the number of divisions per axis and N_{gauss} the number of gaussians.

3.4. Using the algorithm in conjunction with reduced SC-HMM's

When used with reduced SC-HMM's or CD-HMM's, the algorithm poses another problem: for some states, none of the gaussians are evaluated. Removing a promising node from the search beam because of a bad match in a single frame is not a good strategy at all. Therefore we have to estimate the probability of the non evaluated gaussians.

A fixed lower bound on the gaussian probabilities or output probabilities is not an option, as we cannot anticipate the range of these probabilities in advance. Therefore we add a relative lower bound to the gaussian probabilities. This lower bound is estimated as a fraction of the average probability over all gaussians, assuming a zero probability for the non evaluated gaussians. The exact value of the fraction (f) is not critical and was made dependent on the number of gaussians selected per state (N_{sel}) as follows:

$$f = 10^{-\log_2(N_{sel})}$$

3.5. The parameter values and some results

Number of divisions per axis: A good value for the number of divisions for our data bases is 256 to 512. This large value illustrates our statement in section 3.2 about the required number of divisions per axis.

The likelihood-threshold: The value of the threshold is set so that a requested reduction in gaussian-evaluations is achieved. For reduction factors up to 6, no deterioration in recognition results was noticed.

Memory requirements: For 512 divisions per axis, each gaussian index was stored 3.2 times on average per axis. For 256 divisions each index was stored 2.7 times on average. These values are close to the number of parameters needed to describe the gaussians themselves and are considerably below the (approximate) upper bound derived in section 3.3. We also noticed that the memory requirements almost do not depend on the reduction factor.

Overhead of the selection algorithm: The selection algorithm needs about 1/36 of the time needed to evaluate a single gaussian per gaussian in the set. The calculations needed at startup time to create the FRG-info (for 512 divisions per axis) are equivalent to 3000 gaussian evaluations per gaussian in the set.
Note: all timings were calculated for a 25-dimensional input vector.

4. Experimental results

In this section our reduced SC-HMM's are compared with CD-HMM's both on a problem of Isolated Word Recognition (IWR) and in a Continuous Speech Recognition (CSR) experiment.
The main characteristics of all experiments described are:

- Data base: 16 kHz single channel data with SNR ratios between 15 dB and 40 dB (mean 30 dB)
- Signal processing: Mel-cepstrum based (with first and second derivatives, 25 parameters in total), with cepstral mean normalisation
- Context-independent phoneme models for the 44 phonemes of Dutch, and a 1-state noise model
- Both training and recognition are based on Viterbi-alignation

4.1. IWR experiment

Data bases and vocabulary for the IWR experiment:

- The vocabulary of our isolated word training corpus consists of 400 phonetically balanced Dutch words. From each of 134 different speakers, one of 4 subsets of 100 words was recorded. This gives a total of 13377 (long) words for training.
- The data base for testing consists of 20 other Dutch words (the 10 digits and 10 short commands; 13 one-syllable words, 7 two-syllable words) recorded by 40 speakers (other speakers than those in the training data base). This gives 797 utterances in total.
- As vocabulary for the IWR experiment, we added the 2000 most frequent words in Dutch to the 20 data base words, i.e. 2007 different words in total. Of these words, 34% has only one syllable, 44% has two syllables, 22% has three or more syllables.

For the IWR recognition experiments we used 5-state phoneme models. We compare a CD-HMM with 8 gaussians per state (1768 gaussians in total) and a reduced SC-HMM with 32 gaussians per state (1763 gaussians left in total).

In table 1 the results of the CD-HMM and the SC-HMM are compared. In the first column the recognition rate can be found. The other columns give the computational cost for the gaussian probabilities, for the weighted sum in the output probabilities and for the alignments, all of them w.r.t. the cost for the weighted sum in the CD-HMM case (chosen 1.0). For the experiment a beam search algorithm with maximum beam width 1000 was used.

	recog. rate	computational cost		
		gauss.	sum	align.
CD-HMM				
without FRG	70.3%	26.5	1.0	6.9
with FRG	69.9%	5.7	1.0	7.1
SC-HMM				
without FRG	75.3%	26.5	2.2	7.8
with FRG	75.2%	4.6	2.3	7.0

Table 1: Recognition rates and timing on the IWR task

We can see that the FRG-system, set to calculate on the average only 15% of the gaussians, reduces the total cost for calculating the output probabilities (gaussians and weighted sum) with a factor 4 without deterioration in recognition result. And by using our reduced SC-HMM the error rate decreases with almost 20% w.r.t. the CD-HMM.

4.2. CSR experiment

Data bases and vocabulary for the CSR experiment:

- For our training corpus with continuous speech, the 134 training speakers read 5 Dutch paragraphs each. This gives 3197 different train sentences, with 39710 words in total.
- The continuous speech data base for testing consists of 5 paragraphs from each of the 40 test speakers: 981 test sentences with 12469 words in total. Note that the textual descriptions of our continuous data bases were never checked for errors, stuttering, hesitations, breathing, background noises, ...
- As vocabulary for the CSR experiment, we chose all 8543 words in all train and test sentences together. Of these words, 1978 occur in both train and test sentences, 1371 can only be found in the test sentences and 5194 are used in the train sentences solely.

Acoustic modeling in the CSR recognition experiments was done with 3-state phoneme models. We compare a CD-HMM with 16 gaussians per state (2128 gaussians in total) and a reduced SC-HMM with 64 gaussians per state (2074 gaussians left in total). Both models were first trained on the isolated word training data base solely. Then two training passes were added with a corpus that consists of both the isolated word and the continuous speech training data bases.

As for the language modeling, we used the method described in [3] to construct both bi-gram and tri-gram language models for the words in the sentences. At this moment we don't have a large text corpus for Dutch to estimate all language model probabilities. Therefore we first used the set of training sentences as corpus. As 40% of the test sentence words are not available in the training sentences, the perplexity of these language models (on the test set) is high: 550 for the bi-gram and 960 for the tri-gram. To get a CSR experiment with lower perplexity, we also constructed the language models based on all sentences together (train and test sentences). Here we get perplexities of 45 on the bi-gram and 6.5 on the tri-gram.

In table 2 the recognition rate (100% - % insertions - % deletions - % substitutions) is given for all four language models. The results of the CD-HMM and the SC-HMM are compared. For the experiments a beam search algorithm with maximum beam width 5000 was used.

Lang. mod. →	based on train sent.		based on all sent.	
	bi-gram	tri-gram	bi-gram	tri-gram
Perplexity →	550	960	45	6.5
CD-HMM	56.8%	59.4%	79.4%	93.0%
SC-HMM	59.0%	61.4%	82.4%	94.5%

Table 2: Recognition rates on the CSR task

Again the SC-HMM outperforms the CD-HMM in each of the language model cases. As for the IWR experiments, we used the FRG-system in the CSR experiments to evaluate on the average only 15% of the gaussians, again without deterioration in recognition results.

5. Conclusions

In this paper, we described two methods to improve full SC-HMM's. The FRG-system drastically reduces the computational cost for evaluating the set of gaussians, both for CD-HMM's and for SC-HMM's. With the construction of a reduced SC-HMM, the weighted sums in output probability calculations are handled. The resulting models are as fast as CD-HMM's and outperform them in recognition results.
In the future, we will use the reduced SC-HMM to model context dependent phonemes. Then the advantage of modeling the gaussians and the states independently will become even more clear.

6. REFERENCES

1. P. Beyerlein. Fast log-likelihood computation for mixture densities in a high-dimensional feature space. In *Proc. ICSLP*, volume I, pages 271–274, Yokohama, September 1994.
2. X.D. Huang and M.A. Jack. Unified techniques for vector quantisation and hidden Markov modeling using semi-continuous models. In *Proc. ICASSP*, volume I, pages 639–642, Glasgow, April 1989.
3. P. Placeway, R. Schwartz, P. Fung, and L. Nguyen. The estimation of powerful language models from small and large corpora. In *Proc. ICASSP*, volume II, pages 33–36, Minneapolis, April 1993.
4. S. Sagayama and S. Takahashi. On the use of scalar quantization for fast HMM computation. In *Proc. ICASSP*, volume I, pages 213–216, Detroit, May 1995.
5. T. Watanabe, K. Shinoda, K. Takagi, and K. Iso. High speed speech recognition using tree-structured probability density function. In *Proc. ICASSP*, volume I, pages 556–559, Detroit, May 1995.