PROC. OF THE SPEECH PROCESSING WORKSHOP,
MAGDEBURG, 2003

Various Information Sources for HMM with Weighted Multiple Codebooks

Christian Hacker, Georg Stemmer, Stefan Steidl, Elmar Nöth, and
Heinrich Niemann *

Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung, Martensstraße 3,
D-91058 Erlangen, Germany
christian.hacker@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

Abstract. When semicontinuous HMM are used for acoustic modeling in speech recognition usually all states share a single codebook. In our investigations we split the feature set into independent subsets and use separate codebooks for each part. This provides a higher modeling flexibility while keeping the parameter space compact. Further experiments integrate new information sources by using additional codebooks which are estimated in a supervised training. For instance codebooks for phone transitions are applied. Codebook exponents weight the different information sources. Relative reductions in word error rate up to 20 % have been achieved.

1 Introduction

The most common approach in acoustic modeling for speech recognition are hidden Markov models (HMM). Several different types of HMM can be employed. If sufficient training data is available, best recognition accuracy will be achieved with continuous HMM (CHMM). Each HMM state has its own codebook which results in a large amount of parameters that have to be computed. In contrast semicontinuous HMM (SCHMM) have the advantage that training with small amounts of data is more robust due to the smaller number of free parameters. Only one single codebook, which is shared by all HMM states, has to be estimated. State-dependent factors weight the densities of the codebook. Discrete HMM prove to yield poor results, because the vector quantization proceeds in advance and information is lost.

In order to achieve higher precision in acoustic modeling it is possible to increase the number of Gaussian density functions of the codebook. However,

* A part of this work was funded by the European Commission (IST programme) in the framework of the PF-STAR project under Grant IST-2001-37599. The responsibility for the content lies with the authors.

at the same time more free parameters have to be estimated. Usually the word accuracy rises just slightly and converges at a certain maximum.

In the following we present an approach for SCHMM to improve the codebook without reaching the maximum number of free parameters, which would not guarantee a robust training any more. We partition the feature set in a way, that the statistical dependency between the subsets is as low as possible. For each subset an individual codebook is estimated. The output density in each HMM state is a product of densities from different codebooks.

The codebooks are weighted by codebook exponents which are trained on the validation data set. If we retrain the recognizer with the optimal codebook weights, best recognition rates are achieved. Various recognizers with different codebooks and different weightings are evaluated in this paper. Additional information can be found in [2]. In further experiments we extend the approach with additional supervised trained codebooks. Gaussian densities are estimated for phone supersets and phone transitions.

The approach with multiple codebooks is described in a couple of publications in different ways. In [6] the factorized vector quantization speeds up the training process but does not raise the accuracy of the recognizer. In the SPHINX system [4] multiple codebooks are used successfully. However, in the investigations only diagonal covariance matrices are considered. Thus no covariance information is lost by splitting the codebook. Recognition rates are increased with multiple codebooks in [3]. In [5] the data streams are weighted by codebook exponents, that are state dependent and sum up to one. Thereby for different phones either the static or the dynamic characteristics can be emphasized. In [7] the observation of a phone- and a word-recognizer are computed in parallel for each HMM state. Both Gaussian densities are weighted by codebook exponents.

SCHMM with multiple codebooks are explained in Chap. 2. In Chap. 3 the speech recognizer and the database are described. The experimental results are discussed in Chap. 4. The paper ends with conclusion and outlook.

2 SCHMM with Multiple Codebooks

In a standard SCHMM the function $b_j(\boldsymbol{c}_t)$ gives the probability that the feature vector $\boldsymbol{c}_t$ observed at time t is produced by the HMM state j.

$$b_j(\boldsymbol{c}_t) = \sum_{k=1}^{K} c_{jk} P(\boldsymbol{c}_t|k) \tag{1}$$

The features $\boldsymbol{c}_t$ can be generated by any of the K classes of the codebook. The classes k are weighted by the state-dependent class weights c_{jk}.

2.1 Approach with Multiple Codebooks

In the following we assume that the feature vector $\boldsymbol{c}$ has dimension d. For the multiple codebook approach we resort and split $\boldsymbol{c}$ into two or more parts $\boldsymbol{c}_n$. The

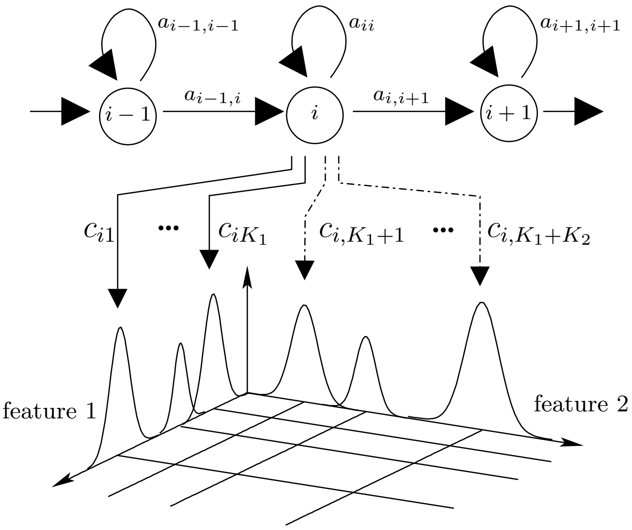

Fig. 1. SCHMM with 1-dimensional codebooks for statistically independent features

statistical dependency between the subsets should be as low as possible. Each of the $\boldsymbol{c}_n$ contains d_n features ($\sum_n d_n = d$). The $\boldsymbol{c}_n$ are called feature streams. The estimation of codebooks for all feature streams is integrated in the Baum-Welch training.

In a simple case we partition the feature vector into $d/2$ static features and $d/2$ dynamic features. Instead of using one codebook with d-dimensional Gaussian densities, we compute two $d/2$-dimensional codebooks. As we assume the feature streams to be independent the probability to observe $\boldsymbol{c}$ in state j can be computed from

$$\begin{aligned} b_j(\boldsymbol{c}) &= b_j(\boldsymbol{c}_{stat}, \boldsymbol{c}_{dyn}) = b_j(\boldsymbol{c}_{stat}) \cdot b_j(\boldsymbol{c}_{dyn}) = \\ &= \left(\sum_{k=1}^{K_1} c_{jk} \cdot P(\boldsymbol{c}_{stat}|k) \right) \cdot \left(\sum_{k=K_1+1}^{K_1+K_2} c_{jk} \cdot P(\boldsymbol{c}_{dyn}|k) \right) \end{aligned} \tag{2}$$

Each sum is over all densities of one of the codebooks. Statistical independence implies uncorrelated features. If the number of classes in each codebook is $K_1 = K_2 = K$, the multiple codebook approach approximates the original $d \times d$ covariance matrices by block matrices of the shape

$$\begin{pmatrix} \begin{matrix} x\ x \\ x\ x \end{matrix} & 0 \\ 0 & \begin{matrix} y\ y \\ y\ y \end{matrix} \end{pmatrix} \tag{3}$$

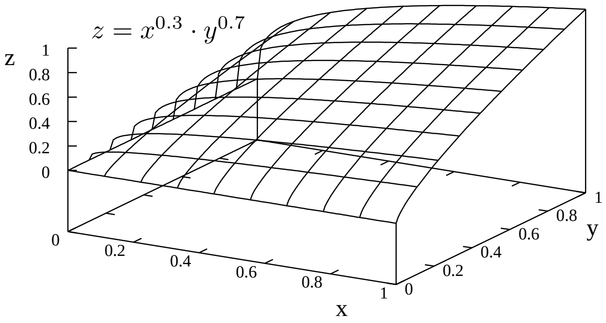

Fig. 2. Product of weighted probabilities $x^{0.3} \cdot y^{0.7}$

where just the $d/2 \times d/2$ squares filled with x or y are used whereas the rest is set to zero. Figure 1 illustrates the situation for $d = 2$. The two features are statistically independent and thus the two one-dimensional feature spaces are drawn orthogonally. The density of the distribution is shown for each feature. For each determined value of feature 1 there is an independent value of feature 2. The probability of all these points "feature 1 *and* feature 2" in the hatched plain is given by the product of the two one-dimensional Gaussian mixture densities. The result is a 2D Gaussian mixture density function covering the plane. Thus the span of two independent one-dimensional codebooks is a two-dimensional one. The covariance between feature 1 and feature 2 is lost. Note, that we get $3 \times 3 = 9$ codebook classes in the two-dimensional feature space in Fig. 1, but only $3 + 3 = 6$ state dependent weights c_{jk}.

In general we split the feature vector in N parts and get N codebooks. Then the probability of $\boldsymbol{c}$ in state j is

$$b_j(\boldsymbol{c}) = \prod_{n=1}^{N} \sum_{k=1}^{K_n} c_{jk} P(\boldsymbol{c}_n|k). \tag{4}$$

The computational effort of the multiple codebook approach is much smaller, as long as the overall number of classes does not exceed the number of classes in the single codebook approach. Otherwise a rather large number of stream-dependent weights c_{jk} has to be computed.

2.2 Codebook Exponents

As in [5] the codebooks are weighted with codebook exponents α_n. This heuristic approach leads to the approximation

$$b_j(\boldsymbol{c}) \approx \prod_{n=1}^{N} \left(\sum_{k=1}^{K_n} c_{jk} P(\boldsymbol{c}_n|k) \right)^{\alpha_n} = \prod_{n=1}^{N} P(\boldsymbol{c}_n|j)^{\alpha_n} \tag{5}$$

We use codebook exponents between 0 and 1 in order to enlarge the probability values. If we set the weight of a codebook to zero, the corresponding part of the feature vector will not be considered as the corresponding probability value $P(\boldsymbol{c}_n|j)$ is set to 1.0. Figure 2 illustrates the product probability $x^{0.3} \cdot y^{0.7}$. The influence of x on the final density value is smaller than y, as the values of $x^{0.3}$ are all closed to 1.0 for $x > 0.2$. The value of $x^{0.3} \cdot y^{0.7}$ is nearly independent of the lower weighted x. Consequently the codebook with the higher exponent has more influence to the product probability and is more important.

2.3 Additional Supervised Trained Codebooks

Above we discussed the problem of generating disjunct parts of the feature set with different codebooks. In further experiments we extend the approach with additional codebooks which all model the complete feature vector $\boldsymbol{c}$. This time processes generating the vector $\boldsymbol{c}$ in state j with different codebooks are assumed to be independent.

The additional codebooks are computed once in a supervised training procedure and are not reestimated any more in the Baum-Welch training. The supervised training is based on a phonetic labeling of the training data. The phone labels are put into six different classes, e.g. vowels, fricatives, ... For each class one Gaussian density is estimated. Other experiments investigate into labels for phone transitions.

3 Speech Database and Training

The system that has been used for the experiments is a speaker independent continuous speech recognizer. For training we use the ISADORA system [6]. The codebook is reestimated 10 times. After each of these reestimation steps the HMM parameters are computed separately in 20 Baum-Welch steps.

For our investigations a part of the German EVAR data set [1] is used. It consists of 7438 utterances, which have been recorded with our conversational train timetable information system. The data is recorded partly via telephone, the rest is band limited through a telephone filter. The total amount of data is approx. 8 hours. 4999 utterances have randomly been selected for training, 441 for validation and 1998 utterances are available for testing.

4 Experimental Results

4.1 Baseline System

Our baseline recognizer uses SCHMM with a codebook of 256 Gaussian densities. The word error rate (WER) is 26.5 %. With 512 codebook classes the WER can be decreased to 25.5 %. Table 1 shows the error rates of the baseline system and specifies the number of parameters that have to be estimated for the symmetric covariance matrices and the mean vectors of all codebook classes. The price for the small improvement of the recognition rate in the second experiment is that the number of free parameters rises from $83,000$ to $166,000$.

Table 1. Baseline recognizer with a single codebook and different classes

Densities per cb	Cb param. $/10^3$	Weights for training	Weights for testing	WER in %
256	83	no weighting		26.5
512	166	no weighting		25.5

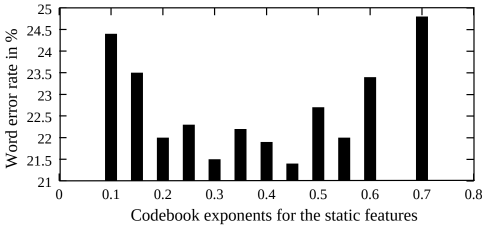

Fig. 3. Word error rate on the validation data set. The codebook exponents for static and dynamic features sum up to 1

4.2 Multiple Codebooks for Different Feature Sets

In first experiments the 12 static features (energy and 11 MFCC) are separated from the 12 dynamic features (derivatives of the static features) and SCHMM with two codebooks are trained. If we use codebooks with 128 classes each, the number of free parameters is $23,000$ and obviously smaller than the one in the baseline recognizer. The number of state dependent weights c_{jk} does not rise either. However, with two unweighted codebooks we have 28.7 % WER. Therefore we evaluate the recognizer on the validation set with different codebook exponents α_n. The results for α_n that sum up to one is shown in Fig. 3. We found minima for $\alpha_{stat} = 0.3$ or $\alpha_{stat} = 0.45$. In a 2D grid search ($\sum_n \alpha_n \neq 1$) another minimum could be found for $\alpha_{stat} = 0.5$ and $\alpha_{dyn} = 0.9$. In the following we remove the constraint that the weights have to sum up to one. For different optima on the validation data set we evaluate the recognizer on the test data set. In the best case the WER drops to 23.5 % (Table 2, rows 1–4). In further investigations the number of codebook classes is varied. With two codebooks, 256 classes each, the WER is decreased to 22.6 % (Table 2, rows 5–7).

Our approach is now to retrain the recognizer with the optimal weights $\alpha_{stat} = 0.3$ and $\alpha_{dyn} = 0.7$ found with the unweighted trained recognizer. Again we look for optimal weights by evaluating the retrained recognizer on the validation data set. On the test set the WER can be reduced to 22.2 % (Table 2, row 8). This is a relative reduction of 16 % in comparison with the baseline system with 256 classes although the number of free codebook parameters is decreased. The number of state dependent weights c_{jk} is in both cases 256 . Thus the size of the HMM file is constant whereas the size of the codebook file is reduced.

In further experiments we put the energy and the derivative of the energy in a separate codebook. Now, the second codebook generates the 11 MFCC and the third one the 11 ΔMFCC. Again the word error rate can be reduced. The

Table 2. SCHMM with two codebooks for 12 static features (cb1) respectively 12 dynamic features (cb2)

	Densities per cb	Cb param. $/10^3$	Weights for training cb1	cb2	Weights for testing cb1	cb2	WER in %
1	128	23	1.00	1.00	1.00	1.00	28.7
2	128	23	1.00	1.00	0.30	0.70	23.5
3	128	23	1.00	1.00	0.50	0.90	24.2
4	128	23	1.00	1.00	0.45	0.55	24.3
5	100	18	1.00	1.00	0.35	0.65	24.2
6	200	36	1.00	1.00	0.30	0.70	22.7
7	256	46	1.00	1.00	0.25	0.75	22.6
8	128	23	0.30	0.70	0.45	0.55	22.2

Table 3. SCHMM with three codebooks for energy (cb1), MFCC (cb2) and ΔMFCC (cb3)

Densities per cb	Cb param. $/10^3$	Weights for training cb1	cb2	cb3	Weights for testing cb1	cb2	cb3	WER in %
256	41	1.00	1.00	1.00	0.30	0.30	0.75	21.6
256	41	0.30	0.30	0.75	0.30	0.30	0.75	21.1

best results are achieved, if we train three codebooks with 256 classes each. The number of codebook parameters is still just half as large as in the baseline system. However, the number of c_{jk} is three times larger. On the validation data set we find the optimal weights for $\alpha_1 = \alpha_2 = 0.3$ and $\alpha_3 = 0.75$. On the test set the word error rate is 21.6 %. After a second training with these optimal weights the word error rate is reduced to 21.1 % (Table 3). The relative improvement of the baseline recognizer is 20 %.

It is possible to use quite more feature streams in order to train SCHMM with a high number of codebooks. Therefore we estimate codebooks for derivatives in three time resolutions. However, the feature streams are highly correlated and we still have a word error rate of 23.0 %.

4.3 Supervised Trained Codebooks

Another way to use SCHMM with multiple codebooks is to estimate different codebooks for the same feature set. The Gaussian density functions for the additional codebook are trained in different ways. First we cluster phone labels that are computed automatically for all feature vectors. We get six phone supersets for vowels, nasales, fricatives, plosives, silence and background noise (configuration 1 in Table 4). We append this codebook as well to the baseline recognizer as to the two codebooks for static and dynamic features. The WER of the baseline system could be lowered, but not the one for the more optimized system with two codebooks (Table 4). In a second approach we add classes for transitions between the six phone supersets and the boundaries of the turn. When two sub-

Table 4. SCHMM with an additional supervised trained codebook. In configuration 1 six phone supersets are used, in configuration 2 additionally superset transitions

Config	Densities	Cb param.	Weights for training			Weights for testing			WER
		$/10^3$	cb1	cb2	cb3	cb1	cb2	cb3	in %
1	256 + 6	85	1.00	1.00	-	0.65	0.30	-	25.2
1	128 + 128 + 6	24	1.00	1.00	1.00	0.20	0.85	0.35	23.4
2	128 + 128 + 44	27	1.00	1.00	1.00	0.20	0.80	0.20	23.0
2	128 + 128 + 44	27	1.00	1.00	1.00	0.20	0.70	0.30	22.5

sequent feature vectors are labeled with different phones, a new transition label is introduced. Transitions, that do not appear very often, are put in the same class and we obtain a codebook with 44 Gaussian densities. The WER can be reduced to 22.5 %. We expect to achieve further improvements if we combine this useful additional information with several experiments described in Chap. 4.2.

5 Conclusion and Outlook

Subject of our investigations was the evaluation of SCHMM with multiple codebooks. The best configuration found is a system with three codebooks for energy, static and dynamic features. We looked for optimal codebook exponents and weighted the third codebook mostly in order to achieve best recognition rates. This appears reasonable because dynamic information is more important for noisy telephone speech. Further improvements could be reached if we retrain the recognizer with weighted codebooks. The word error rate of the baseline system could be reduced by 20 % relative. Additional supervised trained codebooks for phones and phone transitions reduce the error rates as well.

In further work quite more information sources can be utilized and combined. Codebooks for phone transitions seem to be a useful additional information. Other weightings than codebook exponents could reach further improvements. We will look for information sources and weights that are optimal for children's speech.

References

1. Gallwitz, F.; Aretoulaki, M.; Boros, M.; Haas, J.; Harbeck, S.; Huber, R.; Niemann, H.; Nöth, E.: The Erlangen Spoken Dialogue System EVAR: A State–of–the–Art Information Retrieval System. in: Proceedings of 1998 International Symposium on Spoken Dialogue (ISSD 98), Sydney, Australia (1998), pp. 19–26.
2. Hacker, C.: Semikontinuierliche Hidden-Markov-Modelle mit mehreren Kodebüchern. Diploma Thesis (in German), Chair for Pattern Recognition, University of Erlangen-Nuremberg (2002)
3. Huang, X.; Acero, A.; Hon, H.: Spoken Language Processing, pp. 439-441, Prentice Hall (2001)
4. Huang, X.; Lee, K.; Hon, H.; Hwang, M.: Improved Acoustic Modeling with the SPHINX Speech Recognition System: in: Proc. ICASSP '91 (1991), pp. 345–348.

5. Rogina, I.; Waibel, A.: Learning State-Dependent Stream Weights for Multi-Codebook HMM Speech Recognition Systems: in: Proc. Int. Conf. on Acoustics, Speech and Signal Processing, Bd. 1, Adelaide, Australia (1994), pp. 217–220.
6. Schukat-Talamazzini, E. G.: Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen (in German). Vieweg, Braunschweig (1995)
7. Stemmer, G.; Zeissler, V.; Hacker, C.; Nöth, E.; Niemann. H.; A Phone Recognizer Helps to Recognize Words Better. in: Proc. ICASSP '03 (2003), pp. 736–739.