STRUCTURED MARKOV MODELS FOR SPEECH RECOGNITION

F. Wolfertstetter and G. Ruske
Institute for Human-Machine-Communication, Munich University of Technology
Arcisstr. 21, D-80290 München, Germany
E-mail: wol@mmk.e-technik.tu-muenchen.de

ABSTRACT

This paper proposes a new modeling of the structure of speech units as a graph consisting of base functions and a transition network. A cluster algorithm taking into account the actual temporal context of the feature vectors is used to generate the base functions, which are approximated by normal distributions. The subsequent maximum-likelihood training procedure establishes the transition network and adjusts the transition probabilities. The emerging graphs for the speech units are a structure of branching and recombining trajectory segments describing statistical dependencies in the feature vector sequence within the speech units as well as in the transition regions between them. A speaker-independent evaluation shows the superiority of the proposed modeling compared to mixture-state HMMs, even for an equal number of model parameters.

1. MOTIVATION AND PRINCIPLE

The mixture-state Hidden-Markov modeling approach is based on the assumption that speech units consist of a linear sequence of stationary segments, whose probability density can be approximated by a superposition of base functions. Since the emission probability of a vector is conditioned only by its state, preceding and subsequent vectors can influence it only by influencing the vector-state assignment (Viterbi-path). Because of the low number of states and the high number of acoustic events represented by a state, this influence is very weak. Thus, the effective model order is quite low.

To improve this fact, interesting methods have been proposed ([1],[2],[3],[4]) to constrain the probability density functions in the states by actual predecessor feature vectors. In this work we propose a modeling of the temporal structure of phonemes on the level of base functions, which can be associated with acoustic events. We define an acoustic event as a temporal segment of a speech unit uttered by a certain speaker or a homogeneous speaker group in a certain context or in a group of similar contexts. Since articulation is a natural process, normally distributed base functions are used. To generate the base functions, a modified LBG-method taking into account the actual temporal context of the feature vectors is proposed. This algorithm is applied in each phoneme segment assigned to a state of a mixture-state HMM by an initial Viterbi segmentation process. The base functions emerging in two segments are connected fully, if the corresponding states in the segmentation-HMM are connected by a transition. The probabilities of all transitions leaving a base function are initialized by an equal distribution. A Viterbi-based maximum-likelihood training procedure [5] yields graphs for the speech units by eliminating transitions which are never observed and by updating the probabilities of the remaining transitions. These graphs can be called "Structured Markov Models" (SMMs), because they describe the temporal structure of the speech units with its alternative and sequential acoustic events. In [6] an approach is shown, where words are modeled on the level of fenones, where a fenone is a cluster in the feature space. However, the standard LBG-algorithm is used there to generate the base functions and the model for a word consists of a linear sequence of fenones.

2. GENERATION OF BASE FUNCTIONS

The base functions should ideally meet the following requirements, which will be described in the next two subsections: Firstly, we demand to have a base function for each acoustic event whose normal distribution can be separated in the given feature space. Secondly, the base functions should allow the formation of trajectory segments in the training procedure to increase the effective model order.

2.1 Cluster Algorithm

Even the first requirement is not fulfilled very well by the LBG-algorithm [7], since in each iteration it splits *each* cluster, disregarding that some of the clusters already approximate a normal distribution, while others should be splitted more often. Additionally, the number of clusters is not known a priori. For this reasons the LBG-method is used to obtain a high number of initial clusters and, subsequently, the most similar cluster pairs are merged until a *distance threshold* is met. Finally, the vector-prototype assignment is optimized in the same way as in each iteration of the LBG-algorithm. Another algorithm, performing a merge of the most similar cluster pairs after *each* LBG-cluster-doubling step, tended to be instable in some of our experiments.

The cluster-pair similarity is measured by the scale-invariant distance measure m used in [8]. For normal distributions with diagonal covariance matrices we obtain with D denoting the dimension of the feature vector space, μ_{ik} denoting component i of the mean vector of cluster k, c_{ik} denoting its variance, N_k denoting the number of vectors in cluster k and N_{ges} denoting the total number of vectors in all clusters:

$$m = \frac{1}{2 \cdot N_{ges}} \cdot \sum_{i=0}^{D-1} ((N_l + N_r) \cdot \log(c_{ilr}) - N_l \cdot \log(c_{il}) - N_r \cdot \log(c_{ir})) \quad (1)$$

l and r denote two clusters and lr denotes the merged cluster. The merging of a cluster pair is formulated according to [9]:

$$\mu_{ilr} = \frac{N_l}{N_l + N_r} \cdot \mu_{il} + \frac{N_r}{N_l + N_r} \cdot \mu_{ir}$$
$$c_{ilr} = \frac{N_l}{N_l + N_r} \cdot c_{il} + \frac{N_r}{N_l + N_r} \cdot c_{ir} + \frac{N_l}{N_l + N_r} \cdot \frac{N_r}{N_l + N_r} \cdot (\mu_{il} - \mu_{ir})^2 \quad (2)$$

0-7803-2431-5/95 $4.00

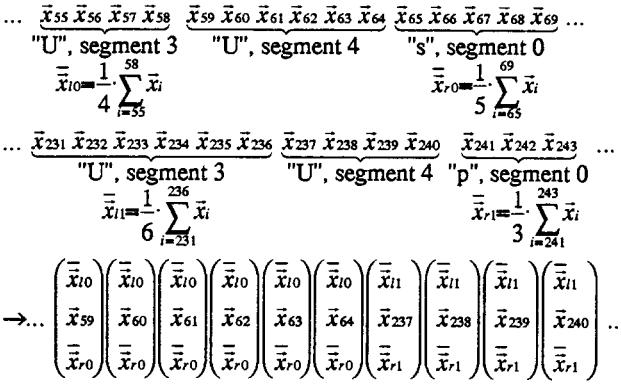

Fig. 1: Calculation of mean vectors in the neighbouring segments of segment 4 of phoneme "U"; the resulting supervectors for the clustering in segment "U", 4 are depicted in the bottom row.

2.2 Context-Dependent Clustering

To meet the second requirement for the base functions, the clustering for each phoneme segment must use information about the actual context of its feature vectors. For this reason, the feature vectors for the clustering are combined using the following scheme:

At each position of the actual phoneme segment in the speech data the temporal mean of the feature vectors assigned to the predecessor segment and the mean of those assigned to the successor segment is calculated. Each vector of the actual segment is then combined with these two mean vectors to a supervector. The representation of the neighbouring segments by their temporal mean is justified, because the segmentation process yields relative homogeneous sections. This is a major difference to linear prediction methods ([1],[2],[3]), which use a fixed predictor offset. Figure 1 illustrates the scheme for segment 4 of phoneme "U". This way, the LBG cluster splitting is controlled by the distances within the actual phoneme segment and the distances in the neighbouring segments. The LBG algorithm will split mainly along the dimensions of the vector space defined by the context features, if there are very different acoustic events in the context. This happens when the actual segment occurs in the neighbourhood of very different segments from different phonemes or in the neighbourhood of very different realizations of one phoneme segment. In the subspace defined by the features from the actual segment, clusters describing the own segment in the context of certain neighbouring events emerge. The base functions can be extracted by isolating the subspace defined by the features from the actual segment. This way, the formation of trajectory segments in the subsequent training procedure is promoted within a temporal span of three segments. Though, it is possible to include more than one neighbouring segment.

Up to now, the features from the actual segment and the context features have the same influence on the clustering. However, a control of the context influence should be possible, because the two requirements for the base functions, good modeling of the acoustic events within a phoneme segment and good context properties, might be contradictory. This can be accomplished by a weighting of the scaling factors for the context features in the LBG-algorithm.

3. RESULTS AND INTERPRETATION

3.1 Experimental Setup

After 16kHz-sampling, windowing in 10ms steps with a 16ms Hamming window and Fast Fourier Transformation 20 spectral channels, zero crossing rate, total energy and two loudness features are extracted within each window. All experiments use the German "Diphon" database containing all possible diphon combinations and a natural distribution of the phoneme a priori probabilities. 7771 sentences from 67 speakers are used for training and 3301 sentences from 33 other speakers are used for evaluation.

3.2 Viterbi and Forward-Backward Processing

For mixture-state HMMs the difference between Viterbi and Forward-Backward (F.-B.) processing is quite low because the summing up of base function emissions in the states is the same in both algorithms. With the base function oriented SMMs, however, Viterbi processing is much more selective. To evaluate this effect, an SMM with 339 base functions was processed by both algorithms. Training and evaluation must be carried out within a phoneme segmentation determined automatically, because the F.-B. algorithm performs no implicit segmentation. Since F.-B. training produces almost no transitions with a probability of zero it does not eliminate very much transitions. Viterbi processing results in a phoneme recognition rate of 54.93%, Forward-Backward processing yields 54.82%. This shows, that Viterbi processing is appropriate for base function oriented models, too. Furthermore, it confirms the theory, that the superposition of base functions in the mixture-states of HMMs is necessary because of the variety of acoustic events represented by a state and not because a single event can't be modeled by a single normal distribution. For this reason, Viterbi processing is used in all other experiments shown in this work.

3.3 Model Structure

Figure 2 visualizes the structure of the SMM for phoneme "U" after the Viterbi-based maximum-likelihood training. The following conclusions can be derived from the evaluation of several SMMs: Firstly, the SMMs only use a small proportion of the transitions modeled implicitly by an equivalent mixture-state HMM and the additional degree of freedom in the transition probabilities is used. For example, the transitions 0→6, 2→6 and 5→6 in SMM "U" obtain the probabilities 0.41, 0.22 and 0.13 while 1→6, 3→6 and 4→6 are never used. A mixture-state HMM would force a common mean value for these transitions.

Secondly, especially consonant-SMMs show one-way transitions between base functions originating from the same phoneme segment. In other words, we obtain a sequential structure within the original segment.

And thirdly, at the model boundaries, groups of neighbouring phonemes and different acoustic forms of the most frequent neighbouring phonemes compete with each other for a specific connection node. The proposed cluster algorithm performs a data driven context definition by yielding specific connection nodes for the most different acoustic events in the context, disregarding whether these differences are caused by speaker variability or by

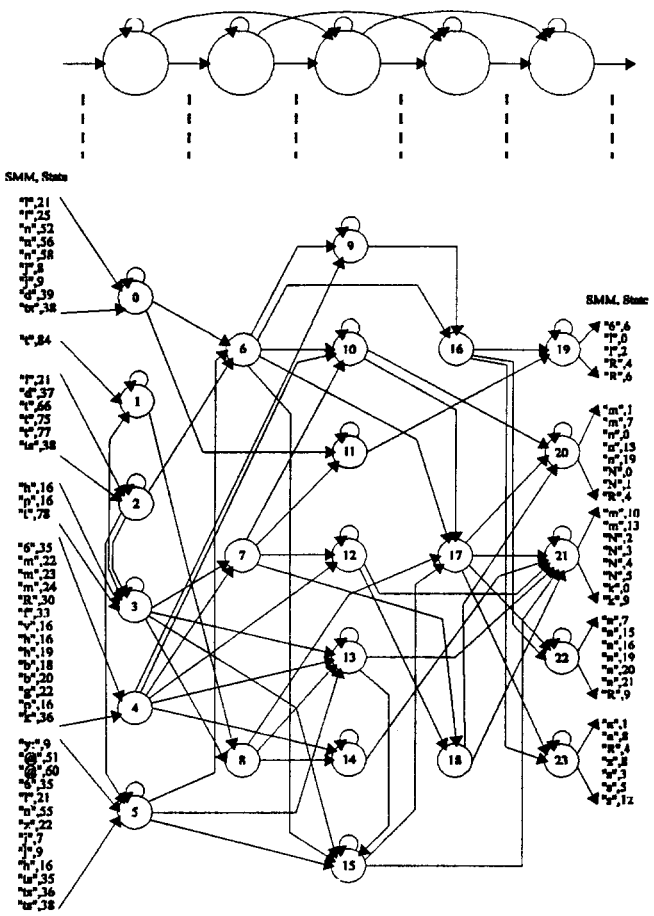

Fig. 2: Segmentation HMM with 5 mixture-states and SMM with 24 states (distance threshold: 0.003, context weight: 0.5)

different phonemes in the context. This is in contrast to context-dependent HMMs (for example [8],[10],[11]).

3.4 Recognition Experiments

In this section results of unconstrained phoneme recognition experiments are shown.

- All results are compared with mixture-state HMMs containing state specific base functions.
- The recognition and insertion rates are determined by a DP-alignment between the recognized phoneme string and the standard transcription. The recognition rate is defined as the number of phonemes recognized correctly divided by the total number of phonemes in the standard transcription and the insertion rate is the number of inserted phonemes divided by this total number of phonemes.
- The recognition rates of the different models are compared over the number of base functions and over the total number of model parameters, because SMMs have more transition parameters than mixture-state HMMs.

In figures 3 and 4, • denotes the results of the mixture-state HMMs, x that of the SMMs without context-dependent clustering and + that of the SMMs with context-dependent clustering using a context weight of 0.5.

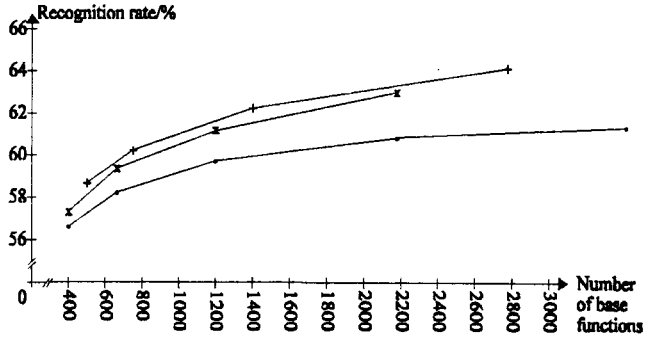

Fig. 3: Recognition rates over number of base functions

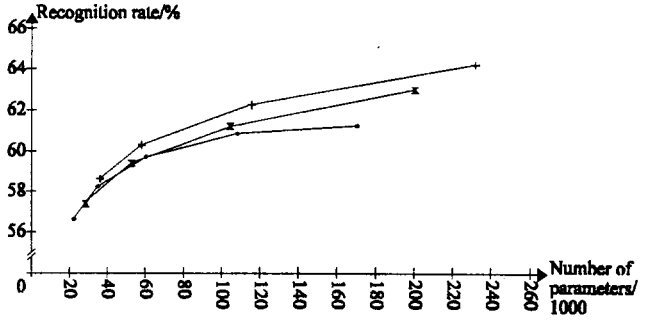

Fig. 4: Recognition rates over total number of model parameters

The comparison at a constant number of base functions as well as at a constant number of parameters shows, that the SMMs with context-dependent clustering yield higher recognition rates than mixture-state HMMs and SMMs without context-dependent clustering. Mixture-state HMMs show significant saturation effects with a rising number of base functions. The SMMs, however, yield an increase of 2 points in the recognition rate with each doubling of the number of base functions in the examined range. For a low number of base functions the SMMs have a worse insertion rate compared to the mixture-state HMMs. However, with the number of base functions rising, this difference decreases to 0.4% (for about 3000 base functions).

Moreover, it is very interesting, that the increase in the number of transitions within and between the SMMs is linearly over the number of base functions and not quadratically. This shows that with a rising number of base functions, mixture-state HMMs model implicitly more and more transitions between base functions which are never observed. With 500 base functions the number of transitions within and between the SMMs is about 50% of the number of implicit base function transitions in the corresponding mixture-state HMMs. With 2776 base functions this value decreases to about 12%.

4. COMPARISON TO TRAJECTORY MODELING

There are 2 basic problems with trajectory modeling. First, one has to choose proper segments for the trajectory clustering. Using segments associated with HMM states [12] or entire phonemes [13],[14],[15] has the consequence that only statistical dependencies within the choosen segments are included. Therefore,

[15] uses "transition tracks" describing the transition between two phonemes. However, the transition region influences the scoring twice, once by the phoneme trajectory and once by the transition track.
The method proposed in this work uses segments associated with HMM-states, but an arbitrary number of neighbouring segments (the *context reach*), which is not limited by phoneme or word boundaries, can influence the clustering. This way, the clustering yields points of a trajectory, each obtaining a normal distribution modeling the error. The combination of this points to a structure of branching and recombining trajectory segments is done by the subsequent data driven training step, which is not influenced by any boundaries, too. This yields trajectory segments within the phonemes as well as trajectory segments describing the transition between different acoustic forms of two phonemes. The tradeoff between the desirable parameter tying resulting from trajectory recombination on the one hand and a high effective model order on the other hand can be controlled by the parameters context weight and context reach. A weakness of the proposed method lies in the separation of trajectory point generation and trajectory structure combination, which causes additional transitions (with a low probability) to and from trajectory points with overlapping error distributions. However, our experiments show that the number of transitions in the models increases linearly over the number of base functions (i. e. trajectory points) and not quadratically. Therefore, this problem becomes less important with the number of base functions rising.
The second problem with trajectory modeling is the time alignment for trajectory clustering. Speaking rate blurs the trajectories without such a procedure [12]. In [13] and [15] all vector sequences are interpolated linearly to obtain a fixed number of trajectory points and [14] proposes a post-training performing a nonlinear alignment against the trajectory points generated by linear interpolation. Nevertheless, the mapping to a fixed number of trajectory points is not optimal.
In the method proposed in this work each trajectory obtains its own number of points automatically, depending on its data vectors. The alignment against the points is nonlinear (Viterbi).

5. DISCUSSION

An optimal representation of all temporal constraints in the speech process could be achieved theoretically by storing an infinite number of feature vector trajectories for large units, for example sentences, and comparing the unknown vector sequences to all these trajectories by DP matching. Of course, this is not possible. The method proposed in this paper represents the reference trajectories as a graph of normally distributed acoustic events. It is based on the assumption, that the position of a feature vector within a normal distribution does not influence the probability of neighbouring vectors. However, the selection of a distribution by the Viterbi algorithm imposes restrictions on the possible neighbouring distributions. This way, each vector influences the probability of its neighbours. The reach of this influence is called the effective model order. It depends on the emerging model structure which in turn depends on the number of context segments and the context weight used in the clustering process, on the number of base functions, and on the data itself.
The most important result of this work is, that a modeling of the structural aspects of the speech units becomes more and more important when the number of base functions rises. This can be seen clearly from the saturation effects in the recognition rate and the high number of incorrect implicit base function transitions in the mixture-state HMM approach. The proposed structured Markov Models are advantageous especially with a high number of base functions.

Acknowledgements

This work was carried out within a project supported by the German Research Foundation („Deutsche Forschungsgemeinschaft", DFG).

References

[1] Maxwell B.A. and Woodland P.C., Hidden Markov Models Using Shared Vector Linear Predictors, Proc. EUROSPEECH 1993, pp. 819-822
[2] Paliwal K.K., Use of Temporal Correlation Between Successive Frames in a Hidden Markov Model Based Speech Recognizer, Proc. ICASSP 1993, pp. 215-218
[3] Woodland P.C., Hidden Markov Models Using Vector Linear Prediction and Discriminative Output Distributions, Proc. ICASSP 1992, pp. 509-512
[4] Takahashi S., Matsuoka T., Minami Y. and Shikano K., Phoneme HMMs Constrained by Frame Correlations, Proc. ICASSP 1993, pp. 219 - 222
[5] Rabiner L.R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989
[6] Bahl, L.R., Brown P.F., de Souza P.V., Mercer R.L. and Picheny M.A., A Method for the Construction of Acoustic Markov Models for Words, IEEE Trans. on Speech and Audio Proc., vol. 1, no. 4, pp. 443-452, 1993
[7] Linde Y., Buzo A. and Gray R., An Algorithm for Vector Quantizer Design, IEEE Trans. on Communications, vol. 28, no. 1, pp. 84-95, Jan. 1980
[8] Bahl L.R., deSouza P.V., Gopalakrishnan P.S., Picheny M.A., Context Dependent Vector Quantization for Continuous Speech Recognition, Proc ICASSP 1993, pp. 632-635
[9] Zhao Y., A Speaker-Independent Continuous Speech Recognition System Using Continuous Mixture Gaussian Density HMM of Phoneme-Sized Units, IEEE Trans. on Speech and Audio Proc., vol. 1, no. 3, pp. 345-361, 1993
[10] Lee K.-F., Hayamizu S., Hon H.-W., Huang C., Swartz J., Weide R., Allophone Clustering for Continuous Speech Recognition, Proc. ICASSP 1990, pp. 749-752
[11] Hwang, M.-Y., Alleva F., Huang X., Senones, Multi-Pass Search, and Unified Stochastic Modeling in SPHINX-II, Proc. EUROSPEECH 1993, pp. 2143-2146
[12] Deng L., Aksmanovic M., Sun X. and Wu C.F.J., Speech Recognition Using Hidden Markov Models with Polynomial Regression Functions as Nonstationary States, IEEE Trans. on Speech and Audio Proc., vol. 2, no. 4, Oct. 1994
[13] Gong Y., Haton J.-P., Stochastic Trajectory Modeling for Speech Recognition, Proc. ICASSP, pp. I-57 - I-60, 1994
[14] Afify M., Gong Y., Haton J.-P., Nonlinear Time Alignment in Stochastic Trajectory Models for Speech Recognition, Proc. ICSLP, pp. S07-27.1 - 3, 1994
[15] Goldenthal W.D. and Glass J.R., Statistical Trajectory Models for Phonetic Recognition, Proc. ICSLP, pp. S31-10.1 - 4, 1994