A SUCCESSIVE STATE SPLITTING ALGORITHM FOR EFFICIENT ALLOPHONE MODELING

Jun-ichi TAKAMI and Shigeki SAGAYAMA

ATR Interpreting Telephony Research Laboratories
Seika-cho, Soraku-gun, Kyoto 619-02, JAPAN

ABSTRACT

In this paper, we propose a new algorithm, Successive State Splitting (SSS), for simultaneously finding an optimal set of phoneme context classes, an optimal topology, and optimal parameters for HMMs commonly using a maximum likelihood criterion. By this algorithm, a Hidden Markov Network (HM-Net), which is an efficient representation of phoneme-context-dependent HMMs, can be generated automatically. We implemented this algorithm and made a preliminary test of this algorithm on the recognition of 6 Japanese consonants (/b/, /d/, /g/, /m/, /n/ and /N/). The HM-Net gave better recognition results with a lower number of total output probability density distributions than conventional phoneme-context-independent mixture Gaussian density HMMs.

INTRODUCTION

In the Hidden Markov Model (HMM) approach to speech recognition, it is a very important issue to generate precise and robust models. To attain precise modeling, it is essential to determine an adequate set of phoneme context classes which well covers the allophonic variations of speech. On the other hand, to attain robust modeling, it is desirable to reduce the total number of model parameters because the number of available training samples is usually limited. These basic ideas lead us to a new efficient framework of phone modeling by shared states where a finite number of hidden states represents probabilistic distributions of acoustic parameters and a state transition network among them covers whole speech variations. This formulation will simultaneously give both precise and robust phoneme-context-dependent phoneme models.

This formulation raises the question of how to determine the following three items:

(1) the model unit, i.e., the set of context classes,
(2) the model architecture, i.e., the number of states per model and the architecture of state sharing,
(3) the model parameters, i.e., output probability density distributions and state transition probabilities.

In conventional approaches, each of these items has been solved independently. For example, item (1) has been solved as an allophone clustering problem[1][2][3], where each phoneme context class is either determined manually using heuristic knowledge or automatically using a distortion criterion. Item (2) has been solved as the shared state problem[4] or the tied output density problem[5] based on a similarity criterion. Item (3) has been solved using the maximum likelihood criterion by the Baum-Welch algorithm.

However, there has been no approach where these items are solved simultaneously as a global optimization problem using the same criterion.

To obtain an approximate solution for this problem, we developed a new algorithm called Successive State Splitting (SSS). In this algorithm, the model unit, the model architecture and the model parameters are determined simultaneously by iteration of state splitting on the contextual domain and temporal domain based on a maximum likelihood criterion, and an efficient network of phoneme-context-dependent HMMs called a Hidden Markov Network (HM-Net) is generated automatically.

In this paper, we will explain the details of the SSS algorithm and the HM-Net, and show the speech recognition experimental results for 6 Japanese consonants (/b/, /d/, /g/, /m/, /n/ and /N/).

THE HIDDEN MARKOV NETWORK

The architecture of an HM-Net is shown in Fig.1. The HM-Net is a network of multiple states, and each state has the following information:

- state number,
- acceptable contextual class,
- lists of preceding states and succeeding states,
- parameters of the output probability density distribution,
- state transition probabilities.

In the HM-Net, if a phoneme context of a sample is given, the model corresponding to the context can be determined by concatenating several states, each of which can accept the context, using the restriction of the preceding state list and the succeeding state list. Since this model

0-7803-0532-9/92 $3.00

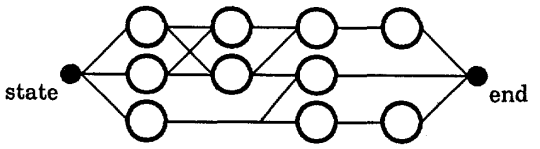

Fig.1 The architecture of the HM-Net

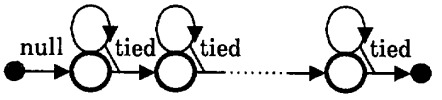

Fig.2 The architecture of a model

and an HMM as shown in Fig.2 are equivalent, we can treat the HM-Net as well as common HMMs.

THE SUCCESSIVE STATE SPLITTING ALGORITHM

Here, we explain the SSS algorithm for automatic generation of the HM-Net.

(1) Step 1 : Training of an initial model

As an initial model, an HMM consisting of one state, S(0), having a diagonal-covariance 2-mixture Gaussian output probability density distribution is trained with all training data containing every phoneme context. A variable M is set to 1.

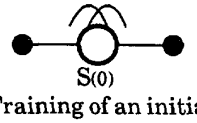

Fig.3 Training of an initial model

(2) Step 2 : Calculation of the distribution size

For each state, a criterion d_i which corresponds to the size of the 2-mixture Gaussian output probability density distribution allocated on S(i) is calculated by Eq.(1),

$$d_i = \sum_{k}^{K} \frac{\sigma_{ik}^2}{\sigma_{Tk}^2} n_i \ , \quad (K\text{: parameter dimension}) \quad (1)$$

where,

$\sigma_{ik}^2 = \lambda_{i1}\sigma_{i1k}^2 + \lambda_{i2}\sigma_{i2k}^2 + \lambda_{i1}\lambda_{i2}(\mu_{i1k} - \mu_{i2k})^2$,
$\lambda_{1i}, \lambda_{2i}$: weight coefficients of state i,
μ_{i1k}, μ_{i2k} : k-th means of of state i,
$\sigma_{i1k}^2, \sigma_{i2k}^2$: k-th variances of state i,
n_i : #training samples for state i,
σ_{Tk}^2 : k-th variance of all samples.

The σ_{ik}^2 in Eq.(1) is equal to the variance expected in the case of applying a single Gaussian output probability density distribution instead of the 2-mixture Gaussian distribution.

The state S(m) having the largest distribution size, d_m, will be split in the next step.

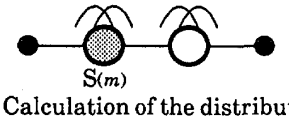

Fig.4 Calculation of the distribution size

(3) Step 3 : Split of the state

The S(m) is split into two states, S'(m) and S(M), each of which has a single Gaussian output probability density distribution corresponding to one of the respective original two Gaussian output probability density distributions. At this time, there are two possibilities on the split domain. One is split on the contextual domain and the other is split on the temporal domain. Therefore, one side of the possibilities which accomplishes a higher likelihood for all the training samples is selected after calculating both the maximum likelihood Pc obtained by split on the contextual domain and the maximum likelihood Pt obtained by a split on the temporal domain. Pc and Pt are calculated as follows, respectively:

• Split on the contextual domain (calculation of Pc)

State splitting on the contextual domain is done by concatenating S'(m) and S(M) in parallel. In this case, since every path allocated on S(m) is also split, all training samples, each of which is accepted on S(m), **Y**, have to be distributed onto one side of two paths, one of which is passing S'(m) and the other of which is passing S(M). Distribution of **Y** is done by splitting the elements belonging to the contextual factor j which accomplishes the maximum value of Pc calculated by Eq.(2),

$$\mathrm{Pc} = \max_{j} \sum_{l} \max \left(\mathrm{p}_m(y_{jl}), \mathrm{p}_M(y_{jl}) \right) \ , \quad (2)$$

where,

j : a contextual factor on S(m),
e_{jl} : an l-th element belonging to the factor j,
y_{jl} : subset of **Y** having the element e_{jl},
$\mathrm{p}_m(y_{jl})$: total likelihood in the case of allocating y_{jl} on the path passing on S'(m),
$\mathrm{p}_M(y_{jl})$: total likelihood in the case of allocating y_{jl} on the path passing on S(M).

After determining the factor j which should be split, the method of distributing e_{jl} is determined by Eq,(3),

$$\begin{cases} e_{jl} \in \mathbf{E}m_j & (\mathrm{p}_m(y_{jl}) \geqq \mathrm{p}_M(y_{jl})) \ , \\ e_{jl} \in \mathbf{E}M_j & (\mathrm{p}_m(y_{jl}) < \mathrm{p}_M(y_{jl})) \ , \end{cases} \quad (3)$$

where,

$\mathbf{E}m_j$: a group of elements which are allocated on the path passing on S'(m),
$\mathbf{E}M_j$: a group of elements which are allocated on the path passing on S(M).

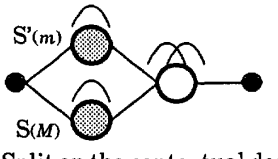

Fig.5 Split on the contextual domain

• Split on the temporal domain (calculation of Pt)

State splitting on the temporal domain is done by concatenating S'(*m*) and S(*M*) in series. In this case, there are two possible models, one of which is allocated S'(*m*) before S(*M*) and the other of which is allocated S(*M*) before S'(*m*). One side of these two possibilities, which accomplishes the maximum value of Pt calculated by Eq.(4), is adopted.

$$Pt = max\left(p_{mM}(Y), p_{Mm}(Y) \right) , \quad (4)$$

where,

$p_{mM}(Y)$: total likelihood in the case of locating S'(*m*) before S(*M*),

$p_{Mm}(Y)$: total likelihood in the case of locating S(*M*) before S'(*m*).

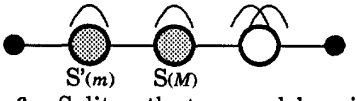

Fig.6 Split on the temporal domain

(4) Step 4 : Retraining of the model

At this time, each of S'(*m*) and S(*M*) has still a single Gaussian output probability density distribution. Then, the model is retrained in order to form a 2-mixture Gaussian output probability density distribution in each of these states and to optimize all states under this condition. After that, S'(*m*) is renamed as S(*m*) and the variable *M* is incremented by 1. the steps from (2) to (4) are repeated until *M* reaches a prescribed number of total states.

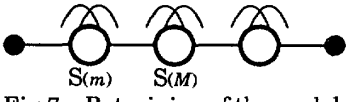

Fig.7 Retraining of the model

(5) Step 5 : Change of distributions

Finally, the HM-Net is retrained to change each output probability density distribution to a voluntary one, e.g., a single Gaussian distribution.

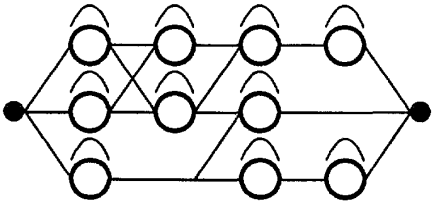

Fig.8 Change of distributions

PHONEME RECOGNITION EXPERIMENTS

To confirm the effectiveness of the SSS, we tested this algorithm on the recognition of 6 Japanese consonants, /b/, /d/, /g/, /m/, /n/ and /N/. For these experiments, we used 6-consonant samples extracted from the following speech data uttered by a native male speaker:

[Training]

•odd-numbered isolated words of the 5240 common Japanese word set (5.68 mora/s),

[Testing]

•even-numbered isolated words of the 5240 common Japanese word set (*word*: 5.68 mora/s),

•conversational sentences uttered phrase by phrase (*phrase*: 7.14 mora/s).

In these experiments, a diagonal-covariance single Gaussian distribution is used as an output probability density distribution of each state. The maximum number of states par model is limited to 4. 34-dimensional coefficients which consist of log-power, 16-channel cepstrum coefficients, delta log-power and 16-channel delta cepstrum coefficients calculated every 5ms were used as feature parameters. In addition, we also tested common phoneme-context-independent diagonal-covariance mixture Gaussian density HMMs having three states for comparison.

Table 1 shows the recognition error rates obtained using the HM-Net, and Table 2 shows the recognition error rates obtained using mixture Gaussian density HMMs. In these tables, ND means the number of total Gaussian output probability density distributions. In the HM-Net, ND is equal to the total number of states, and in the mixture Gaussian density HMM, ND is calculated by multiplying the number of models, 6,

Table 1 Recognition error rates for 6 consonants using the HM-Net.

#states	ND	error rates for top 1. (top 3) (%)	
		word	*phrase*
30	30	9.0 (0.39)	27.9 (6.1)
50	50	5.5 (0.25)	26.6 (6.2)
70	70	5.0 (0.34)	24.2 (7.8)
90	90	4.6 (0.39)	23.6 (7.7)
110	110	3.7 (0.34)	24.1 (7.7)
130	130	4.1 (0.34)	24.7 (8.2)
150	150	3.6 (0.49)	26.1 (9.9)
200	200	2.6 (0.44)	24.9 (12.2)

Table 2 Recognition error rates for 6 consonants using mixture Gaussian density HMMs.

#mixtures	ND	error rates for top 1. (top 3) (%)	
		word	*phrase*
1	18	13.5 (0.44)	26.5 (5.8)
3	54	6.7 (0.15)	25.6 (7.8)
5	90	4.6 (0.15)	23.9 (7.9)
10	180	3.8 (0.05)	25.1 (9.1)
15	270	2.8 (0.05)	26.7 (8.6)

by the number of states per model, 3, by the number of mixtures per state. From these results, it was found that the HM-Net gave better recognition results with a lower number of total output probability density distributions than phoneme-context-independent mixture Gaussian density HMMs.

AN ILLUSTRATION OF THE HM-NET

Fig.9 shows an illustration of the HM-Net having 30 states. In this figure, one oval shows one state, and 4 lines in each oval respectively show the state number, list of preceding phonemes, list of center phonemes, and list of succeeding phonemes. Moreover, ".." written in the list shows that several following phonemes are abbreviated. The acceptable phoneme context class of the state can be defined as all direct products (all combinations) of the preceding phonemes, the center phonemes and the succeeding phonemes.

From this figure, it is found that the front states, e.g., #3, #27, #9, #26 and #15, tend to be split under the influence of the preceding phoneme and the rear states, e.g., #11, #17, #1 and #23, tend to be split under the influence of the succeeding phoneme with the exception of #10 and #28. This tendency seems to be reasonable[1].

From this, we confirmed that contextual variations in the phoneme segment are well reflected in the architecture of the HM-Net.

CONCLUSIONS

We proposed a new algorithm, Successive State Splitting (SSS), for simultaneously finding an optimal set of phoneme context classes, an optimal topology, and optimal parameters for HMMs commonly using a maximum likelihood criterion. We implemented this algorithm and made a preliminary test of this algorithm on the recognition of 6 Japanese consonants (/b/, /d/, /g/, /m/, /n/ and /N/). The HM-Net gave better recognition results with a lower number of total output probability density distributions than conventional phoneme-context-independent mixture Gaussian density HMMs.

Our future research topics are as follows:

- generation of the HM-Net for all Japanese phonemes and its evaluation,
- expansion to continuous speech recognition by combination with the context-dependent LR-parser[6],
- application of the HM-Net to speaker adaptation.

ACKNOWLEDGMENTS

We would like to thank Dr.A.Kurematsu, President, ATR Interpreting Telephony Research Laboratories, for his continuous support of this work. We also acknowledge all the members of the Speech Processing Department for their discussion and encouragement.

REFERENCES

[1] R.Schwartz et al., "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech," ICASSP'85, 31.S3.1.

[2] S.Sagayama, "Phoneme Environment Clustering for Speech Recognition," ICASSP'89, 24.S8.3.

[3] K.F.Lee et al., "Allophone Clustering for Continuous Speech Recognition," ICASSP'90, 53.S14.8.

[4] X.D.Huang et al., "Improved Acoustic Modeling with the SPHINX Speech Recognition System," ICASSP'91, 10.S5.24.

[5] S.Euler et al., "Extending The Vocabulary of a Speaker Independent Recognition System," ICASSP'91, 10.S5.13.

[6] A.Nagai et al., "Phoneme-context-dependent LR Parsing Algorithms for HMM-based Continuous Speech Recognition," Eurospeech'91, S48.3.

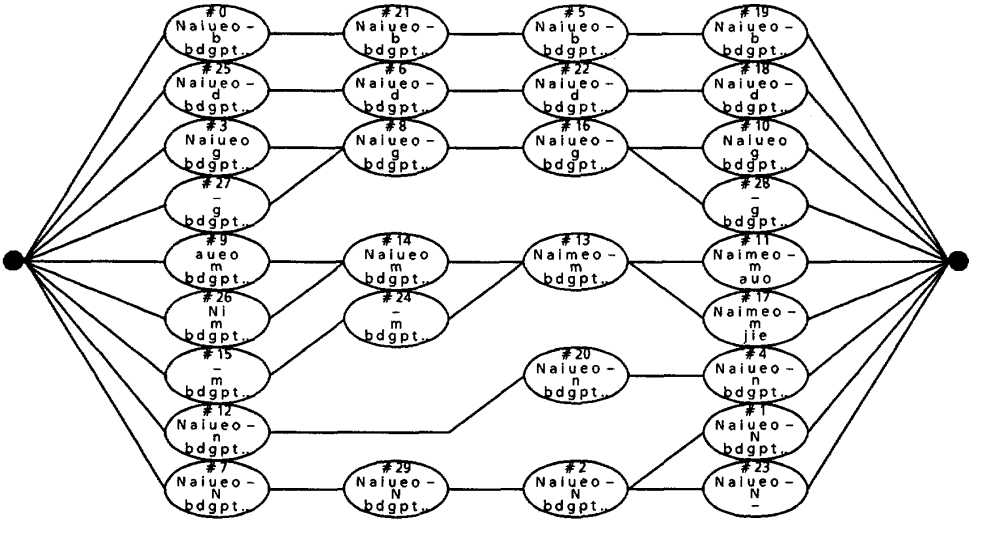

Fig.9 An illustration of the HM-Net for 6 Japanese consonants (#states: 30)