SPEAKER ADAPTATION USING TREE STRUCTURED SHARED-STATE HMMS

Jun Ishii [1] *Masahiro Tonomura* [1] *Shoichi Matsunaga* [2]

[1]ATR Interpreting Telecommunications Research Labs.
2–2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
[2]NTT Human Interface Labs.
1–2356 Take, Yokosuka, Kanagawa 238-03 Japan

ABSTRACT

This paper proposes a novel speaker adaptation method that flexibly controls state-sharing of HMMs according to the amount of adaptation data. In our scheme, acoustic modeling is combined with adaptation to efficiently utilize the acoustic models sharing characteristics for adaptation. The shared-state set of HMMs is determined by using tree-structured shared-state HMMs created from the history recorded for acoustic model generation. The proposed method is applied to the parameter-tying and parameter-smoothing techniques. Experiments have been performed on a Japanese phoneme recognition test using continuous density mixture Gaussian HMMs. Using 50 adaptation phrases, a 42% reduction in the phoneme recognition error rate from the speaker-independent model was achieved.

1. INTRODUCTION

To achieve practical use of speech recognition for many applications, speaker-independent (SI) speech recognition systems with continuous mixture density HMMs (CDHMM) have recently been developed. The SI model consists of many parameters that are trained with a large amount of data, which is necessary to express the speech variations of many speakers. The performance of the SI model, however, is still poorer than that of a well trained speaker-dependent (SD) model.

Speaker adaptation can be utilized, to reduce the performance gap. As the SI model has a large number of parameters, a large amount of adaptation data is needed to obtain stable adaptation performance [1][2]. However, the amount of available speaker-specific data is generally limited. Accordingly, devising a way to share and use correlations of parameters, states or models is one of the key tasks for obtaining rapid and robust adaptation with a small amount of adaptation data.

To effectively carry out speaker adaptation with a small amount of data, the parameter-tying and parameter-smoothing techniques has been proposed [3]. The performance by this method becomes saturated when adaptation data increases. To keep high performance for a wide range of adaptation data, a clustering tree structure of the parameters has recently been adopted for parameter-sharing [4][5]. However, the clustering criterion of adaptation is different from acoustic modeling.

This paper presents a scheme that encompasses acoustic modeling and adaptation. In our scheme, a state-sharing mechanism is included in the modeling algorithm. The successive state splitting (SSS) modeling algorithm [6] is investigated for this strategy. The SSS process gradually expands one state model to a larger Hidden Markov Network (HMnet) that represents all context-dependent phoneme HMMs and maintains state sharing among different phones. Depending on the SSS splitting history, a tree structure (SSS-Tree-Structure) that expresses hierarchical similarities among all phoneme states is created. Using a part of this tree structure, the state-sharing is carried out according to the amount of adaptation data.

The following section describes the procedure of the SSS-Tree-Structure and outlines of the state-sharing method using the SSS-Tree-Structure. Next, an application of the SSS-Tree-Structure to the parameter-tying and parameter-smoothing techniques is shown. Finally, the experimental results are provided on a Japanese phrase data.

2. ADAPTATION METHOD

2.1. SSS-Tree-Structure

Successive state splitting (SSS) is a useful algorithm for HMM topology design. The SSS generates a network of HMM states (HMnet) that can be increased in size by choosing to split the state with the most variability and then determining the splitting domain (either temporal or contextual) for that state. The iteration of this splitting results in an HMnet that efficiently represents contextual and temporal variability of specified sub-word units.

Figure 1 shows an example of the top-down state splitting process for modeling and the corresponding tree structure. First, state 0, which is the initial model, is split into two states, i.e., state 0 and state 1. This is done by the SSS algorithm and the process is described as an SSS-Tree-Structure whose parent node has a unsplit state and whose child nodes have split states. Next, state 1, the state with the most variability, is split into state 1 and state 2, so that state 1

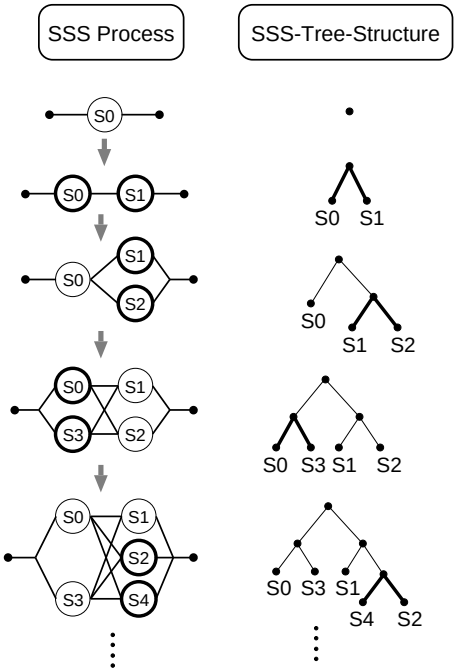

Figure 1: SSS process and SSS-Tree-Structure

and state 2 structure the child nodes. Thus, the SSS-Tree-Structure is created according to an iterative SSS splitting process until the number of states reaches a prescribed number (e.g. 200). This procedure simultaneously generates the HMnet, which effectively represents the context dependent models with small states, and the SSS-Tree-Structure, which assures good performance for a wide range of adaptation data.

The SSS-Tree-Structure allows the shared-state set to be controlled according to the amount of adaptation data to obtain efficient adaptation performance. In order to increase the state-sharing count when the amount of data is small, the shared-state set contains states that are leaf node states under the high-level parent node. On the other hand, in order to decrease the state-sharing count, when the amount of data is sufficient, the shared-state set contains states that are leaf node states under the low-level parent node.

In this SSS-Tree-Structure, since some states that located under the node were once one-state models in the SSS process, the acoustic phenomena of these states are similar, and this similarity is higher at a lower node. Rapid and robust adaptation is expected, as the state-sharing is controlled by the similarity of the state acoustic phenomena.

2.2. MAP-VFS

Vector Field Smoothing (VFS), which encompasses the parameter-tying and parameter-smoothing techniques, has been proposed to deal with untrained or insufficiently trained parameters [3]. VFS consists of three steps: (1) estimation of transfer vector , (2) interpolation and (3) smoothing. The transfer vectors imply a difference between the mean vectors

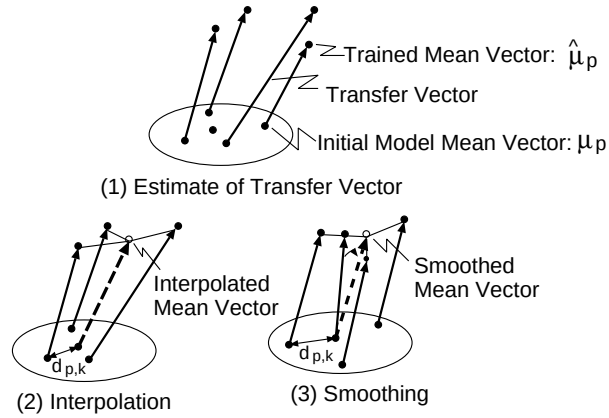

Figure 2: Three procedural constituents of Vector Field Smoothing (VFS)

of a Gaussian distribution for the initial model and those for the target model. Figure 2 shows a schematic view of these procedures.

The mean vector of the initial model is recalculated through embedded training. Each mean $\hat{\mu}_p$ is estimated by maximum likelihood estimation or MAP [7]. In this paper, we adapt MAP estimation for VFS (MAP-VFS) [8][9]:

$$\hat{\mu}_p = \frac{n_p}{n_p + \tau} m_p + \frac{\tau}{n_p + \tau} \mu_p, \tag{1}$$

where, μ_p is the p-th mean vector of the Gaussian distribution of the initial model, m_p denotes the maximum likelihood estimate, n_p denotes the total number of training samples observed for the corresponding Gaussian mixture component, and τ indicates the relative balance between *a prior* knowledge and empirical data. The transfer vector v_p is represented as follows:

$$v_p = \hat{\mu}_p - \mu_p. \tag{2}$$

Since there is a limited amount of training data, there are untrained mean vectors, and the reliability of the retrained mean vectors is poor. Thus, interpolation of the untrained mean vectors and smoothing of the trained mean vectors are performed:

$$\tilde{\mu}_p = \frac{\sum_{k \in N(p)} \lambda_{p,k} v_k}{\sum_{k \in N(p)} \lambda_{p,k}} + \mu_p, \tag{3}$$

$$\lambda_{p,k} = \exp\left(\frac{-d_{p,k}}{f}\right), \tag{4}$$

where, $\tilde{\mu}_p$ is the estimate of the p-th mean vector. $N(p)$ is the set of K-nearest neighbor mean vectors to μ_p, $\lambda_{p,k}$ is the weighting coefficient that denotes the distance $d_{p,k}$ between μ_p and μ_k, and f is a weight control parameter.

In conventional VFS, the K-nearest neighbor mean vectors contained in $N(p)$ that are shared-parameter sets are selected in the order of distances $d_{p,k}$. This is not considered suitable

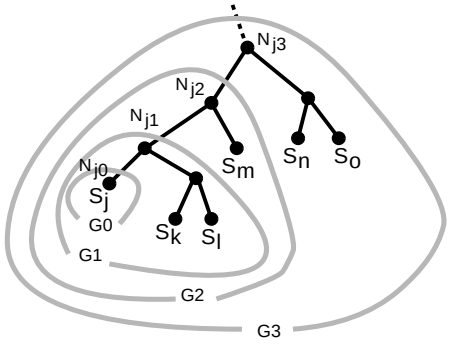

Figure 3: State-sharing area using SSS-Tree-Structure

tying as it dose not take into consideration the similarities of acoustic phenomena. In our scheme, to achieve efficient parameter-sharing, the nearest neighbor mean vectors are selected by using the SSS-Tree-Structure that constitutes hierarchical similarities of parameters.

2.3. MAP-VFS using SSS-Tree-Structure

This section describes parameter-sharing of MAP-VFS using the SSS-Tree-Structure. Let K be the number of trained mean vectors for interpolating and smoothing, μ_p be the target mean vector in the initial model, and $\hat{\mu}_k$ be the trained mean vector of μ_k. For a multivariate Gaussian mixture, the set of K-neighbor mean vectors $N(p)$ is formed as follows.

1. Search the leaf node state including μ_p in the tree.
2. Determine the cluster.
   Go up to the parent node until the total number of trained mean vectors $\hat{\mu}_k$ under this node becomes larger than K.
3. Select the K-trained mean vector in the cluster.
   Select K-trained mean vectors $\{\hat{\mu}_k\}$ according to the distance between μ_p and μ_k.

After this procedure, interpolation and smoothing are performed using transfer vectors $\{v_k\}$ ($\{\hat{\mu}_k - \mu_k\}$).

A simple example is shown in Figure 3. The example assumes that state S_j contains mean vector μ_p. The first selection area is G_0. If the number of trained mean vectors is smaller than K, then the parent node N_{j1} is the node to go up to. Until the number of trained mean vectors is larger than K, it is necessary to go up the nodes repeatedly, e.g. $N_{j1} \rightarrow N_{j2} \rightarrow N_{j3}$, thus extending the selection area ($G_1 \rightarrow G_2 \rightarrow G_3$).

3. EXPERIMENTS

3.1. Conditions

Figure 4 shows a block diagram of the proposed adaptation procedure using the SSS-Tree-Structure. The experimental

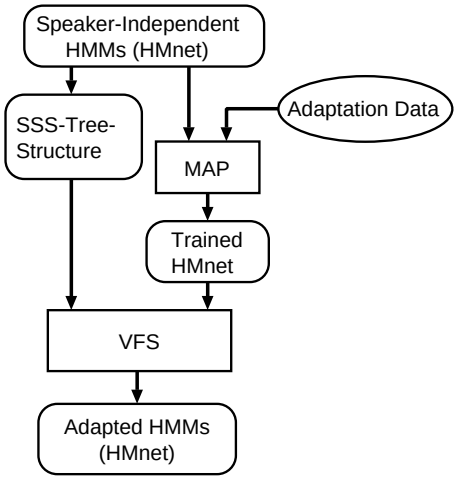

Figure 4: Adaptation procedure

Table 1: Experimental conditions

Analysis conditions	
Sampling frequency 12 kHz Hamming window 20 ms, Frame period 5 ms	
Feature parameters	
16-order LPC-Cepstrum + 16-order ΔLPC-Cepstrum + log power + Δlog power	
Topology	
200-state HMnet (5 mixture per state) trained using 2620 words of one speaker	
Training data of the SI model	
146 males + 139 females (50 Japanese sentences per person)	
Adaptation/Recognition data	
Speakers	4 males and 3 females
Adaptation	N phrases from 598 Japanese phrases
Recognition	279 Japanese phrases

conditions are listed in Table 1. The topology of the HMnet was generated with SSS algorithm using 2,620 isolated words of one male speaker; the SSS-Tree-Structure was also created. The number of states in the HMnet was 200. Diagonal covariance matrices were used and the number of mixture components per state was 5. The SI model was generated [10] by using 50 utterances of 285 speakers.

The proposed algorithm was evaluated using a Japanese 26-phoneme recognition test. For the test, 279 phrases were uttered by 7 speakers who were not included in the SI model training. The adaptation training data was sampled from 598 phrases that differed from test phrases. Giving consideration to the dependence on training data for the speaker adaptation performance, the experiment was repeated three times with different training data. Supervised adaptation was performed only for the mean vectors of Gaussian output distributions. The number of trained mean vectors K described in 2.3 was set to 6.

3.2. Results

Figure 5 shows phoneme recognition error rates of MAP-VFS with the SSS-Tree-Structure while varying the number

of adaptation phrases. τ of the MAP estimation in Eq. (1) is 4.0 for all mean vectors. As a comparison, we plotted the results of conventional VFS and MAP-VFS, whose K-neighbor vectors are selected without the SSS-Tree-Structure.

The performance of MAP-VFS with the SSS-Tree-Structure is superior to that of conventional VFS. The proposed method is slightly inferior to conventional MAP-VFS when there is a small amount of adaptation data (number of adaptation phrases less than 10).

The reason for this was considered as follows. Figure 6 shows the ratio of the number of mean vectors going up levels of a tree structure from the leaf node to the total mean vectors. When the amount of data is small, the vectors whose going up level exceed 3 are substantially contained. The shared-state sets are constituted by the states that are leaf nodes under a high-level parent node. Thus, the similarity of state acoustic phenomena is poorly reflected.

MAP-VFS with the SSS-Tree-Structure achieves a higher recognition performance than dose conventional MAP-VFS for a large amount of adaptation data (number of adaptation phrases more than 20), demonstrating the effectiveness of our approach. When the amount of data is large, going up level from the leaf nodes occupies 1 level up and 2 level up (see Figure 6, over 20 phrases). The shared-state sets are constituted by the states that are leaf nodes under lower-level parent nodes. Furthermore, this approach makes good use of the similarity of state acoustic phenomena.

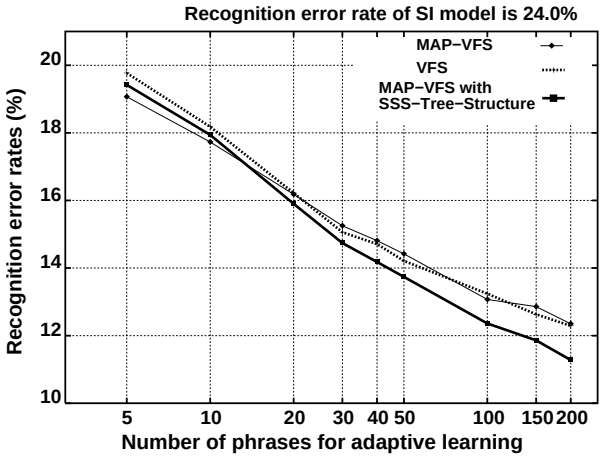

Figure 5: Results of recognition error rates while varying the number of adaptation phrases (average of 7 speakers)

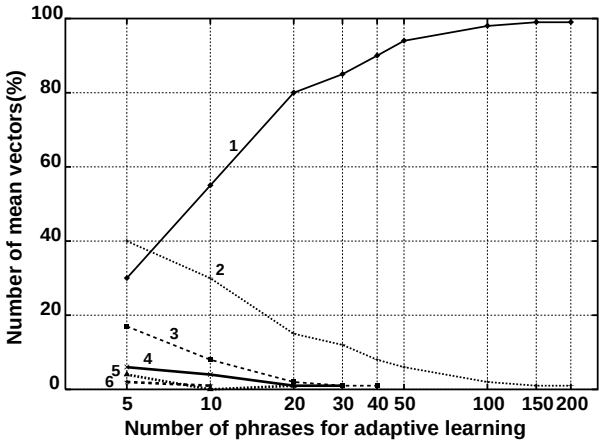

Figure 6: Ratio of the number of mean vector going up levels from the leaf nodes of a tree structure while varying the number of adaptation phrases

4. CONCLUSION

This paper proposed a shared-state speaker adaptation method using an SSS-Tree-Structure. The shared-state set of HMMs are determined using the history of acoustic model generation created by the SSS algorithm. In our scheme, acoustic modeling is combined with adaptation to make the best use of acoustic model tying characteristics. The proposed method is effective when applied to MAP-VFS. Experimental results on a Japanese phoneme recognition test show that the proposed method yields higher recognition performance than dose conventional MAP-VFS when over 20 adaptation phrases are used.

REFERENCES

[1] C. J. Leggetter and P. C. Woodland: "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171-185, 1995.

[2] V. Digalakis and L. Neumeyer: "Speaker Adaptation Using Combined Transformation and Baysian Method," *Proc. of ICASSP95*, pp. 680-683, 1995.

[3] K. Ohkura, M. Sugiyama and S. Sagayama: "Speaker Adaptation Based on Transfer Vector Field Smoothing with Continuous Mixture Density HMMs," *Proc. of ICSLP92*, pp. 369-372, 1992.

[4] C. J. Leggetter and P. C. Woodland: "Flexible Speaker Adaptation For Large Vocabulary Speech Recognition," *Proc. of EUROSPEECH95*, pp. 1155-1158, 1995.

[5] K. Shinoda and T. Watanabe: "Speaker Adaptation with Autonomous Control Using Tree Structure," *Proc. of EUROSPEECH95*, pp. 1143-1146, 1995.

[6] J. Takami and S. Sagayama: "A Successive State Splitting Algorithm for Efficient Allophone Modeling," *Proc. of ICASSP92*, pp. 573-576, 1992.

[7] C.-H. Lee, C.-H. Lin and B.-H. Juang: "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models," *IEEE Trans. on Signal Processing*, vol. 39, no. 4, pp. 806-814, 1991.

[8] J. Takahashi and S. Sagayama: "Telephone Line Characteristic Adaptation Using Vector Field Smoothing Technique," *Proc. of ICSLP94*, pp. 991-993, 1994.

[9] M. Tonomura, T. Kosaka and S. Matsunaga: "Speaker Adaptation Based on Transfer Vector Field Smoothing Using Maximum a Posteriori Probability Estimation," *Proc. of ICASSP95*, pp. 688-691 1995.

[10] T. Kosaka, S. Matsunaga and M. Kuraoka: "Speaker-Independent Phone Modeling Based on Speaker-Dependent HMMs," *Proc. of ICASSP95*, pp. 441-444, 1995.