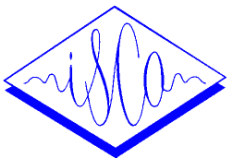

ISCA & IEEE Workshop on Spontaneous
Speech Processing and Recognition
Tokyo Institute of Technology, Tokyo, Japan
April 13–16, 2003

A SUCCESSIVE STATE SPLITTING ALGORITHM BASED ON THE MDL CRITERION BY DATA-DRIVEN AND DECISION TREE CLUSTERING

Takatoshi Jitsuhiro, Tomoko Matsui, Satoshi Nakamura

Spoken Language Translation Research Laboratories,
Advanced Telecommunications Research Institute International,
2-2-2 Hikaridai, "Keihanna Science City", Kyoto 619-0288, Japan.
{takatoshi.jitsuhiro, tomoko.matsui, satoshi.nakamura}@atr.co.jp

ABSTRACT

We propose a new Successive State Splitting (SSS) algorithm based on the Minimum Description Length (MDL) criterion to design tied-state HMM topologies automatically. The SSS algorithm is a mechanism for creating both temporal and contextual variations based on the Maximum Likelihood (ML) criterion. However, it also needs to empirically predetermine control parameters for use as stop criteria, for example, the total number of states. We introduce the MDL criterion to the ML-SSS algorithm so that it can automatically create proper topologies without such parameters. Experimental results show that our extended algorithm can automatically stop splitting and obtain more appropriate HMM topologies than the original one. We also extend the MDL-SSS algorithm by using phonetic decision tree clustering for contextual splitting. A method using a combination of phonetic decision tree clustering and data-driven clustering can automatically obtain almost the same performance as the original method.

1. INTRODUCTION

Phonetic decision tree clustering[1] was proposed as a method of generating tied-state structures of acoustic models for speech recognition. Methods based on phonetic decision clustering originally used the Maximum Likelihood (ML) criterion to choose the phonetic question with which each state was split. However, owing to the nature of the ML estimation, the likelihood value for training data increases as the number of parameters increases. Consequently, it is impossible to stop splitting using only the ML criterion. The methods based on the ML criterion require heuristic stop criteria, such as the total number of states.

Recently, information criteria such as the Minimum Description Length (MDL) have been introduced as splitting and stop criteria in context dependent HMM creation using phonetic decision tree clustering[2]. These methods continue to split states so as to improve the information criteria.

The Successive State Splitting (SSS) algorithm was originally proposed to create a network of HMM states of speaker dependent models[3] and was subsequently expanded to the ML-SSS algorithm for speaker independent models[4]. The ML-SSS algorithm has the same problem as phonetic decision tree clustering in that it requires the total number of states as the stop criterion. The ML-SSS algorithm is a bottom-up approach that conducts both contextual clustering and temporal splitting. The maximum number of temporal states for each phoneme model should be given as another stop criterion for temporal splitting. It is difficult to properly preset these two stop criteria.

We propose an HMM topology design method using the ML-SSS algorithm in conjunction with the MDL criterion as the splitting and stop criteria. We call the new method the MDL-SSS algorithm. The MDL criterion was successfully introduced in phonetic decision clustering as the criterion of contextual clustering[2]. This paper extensively uses the MDL criterion as the criterion for both contextual and temporal splitting in the ML-SSS algorithm.

In Section 2, we explain the ML-SSS algorithm and the stop-splitting problem. The MDL criterion is described in Section 3. We define the MDL-SSS algorithm in Section 4. In Section 5, we evaluate the performance of MDL-SSS. Additionally, phonetic decision tree clustering is introduced into MDL-SSS in Section 6. We summarize our findings in Section 7.

2. ML-SSS ALGORITHM

2.1. Problems of ML-SSS

The ML-SSS algorithm has contextual and temporal splitting[4]. Figure 1 shows the flow of ML-SSS. First, both contextual and temporal splitting are performed for all states. Second, the gains of both contextual and temporal splitting are calculated. Finally, these expected gains are compared with each other and the state with the best gain among all states is selected.

ML-SSS needs the total number of states, N_s, and the maximum length of state sequences for phoneme models, N_p. These parameters must be given before starting the splitting. For temporal splitting, ML-SSS creates one more state and connects it to the original state. The parameters of two distributions are estimated by the forward-backward algorithm, and the total expected gain of temporal splitting is also calculated for the temporal split states. Since it is costly to re-estimate the parameters of all states at every splitting, only the parameters for the two candidate states are re-estimated by using probabilities weighted by the statistics of the target state. Therefore, the likelihood value of temporal split states is an approximate value and it is difficult to use it as a stop criterion. So, N_p is needed as a stop criterion. It is difficult to find the optimal values of these parameters, N_s and N_p. Some experiments should be done for several combinations of the values.

2.2. Gain function by ML-SSS algorithm

Next, we describe the total expected gains of splitting states with the ML-SSS algorithm. The total expected gain of contextual split-

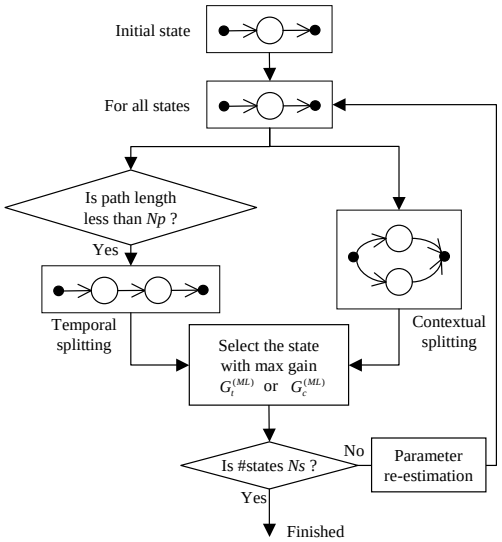

Fig. 1. Flow chart of ML-SSS.

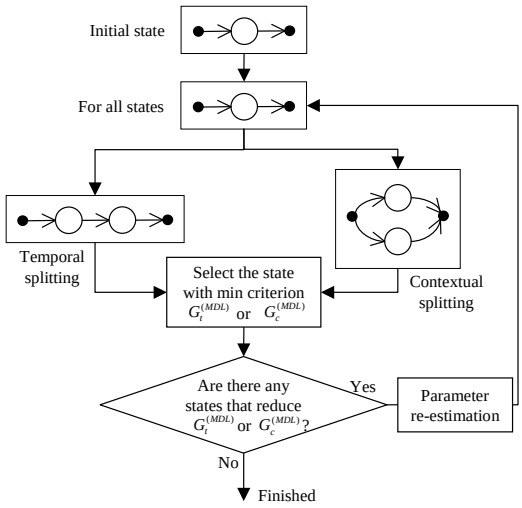

Fig. 2. Flow chart of MDL-SSS.

ting for state S_i split into two new states S_{i_1} and S_{i_2} is

$$G(S_i) = G_{output}(S_i) + G_{trans}(S_i), \tag{1}$$

where $G_{output}(S_i)$ is the difference of output probabilities and $G_{trans}(S_i)$ is the difference of transition probabilities.

$$\begin{aligned} G_{output}(S_i) &= -\frac{1}{2}\{\Gamma(S_{i_1})\log|\Sigma(S_{i_1})| \\ &\quad + \Gamma(S_{i_2})\log|\Sigma(S_{i_2})| - \Gamma(S_i)\log|\Sigma(S_i)|\}, \end{aligned} \tag{2}$$

$$\begin{aligned} G_{trans}(S_i) &= \Xi(S_{i_1}, S_{i_1})\log a_{i_1 i_1} \\ &\quad + \{\Gamma(S_{i_1}) - \Xi(S_{i_1}, S_{i_1})\}\log(1 - a_{i_1 i_1}) \\ &\quad + \Xi(S_{i_2}, S_{i_2})\log a_{i_2 i_2} - \Xi(S_i, S_i)\log a_{ii}, \end{aligned} \tag{3}$$

where $\Gamma(S_i) = \sum_t \gamma_t(S_i)$ is the expected frequency of transition from state S_i. $\gamma_t(S_i)$ is the probability of staying in S_i at the time t. $\Xi(S_i, S_j) = \sum_t \xi_t(S_i, S_j)$ is the expected frequency of transition from S_i to S_j. $\xi_t(S_i, S_j)$ is the probability of transition from S_i to S_j at t. a_{ii} is the self-loop probability.

For contextual splitting, since the transition probabilities are not re-estimated to reduce the amount of calculation, the total expected gain related to only the observation distributions is calculated. For temporal splitting, the transition probabilities are considered because one transition probability is created after temporal splitting. The splitting conditions $G_c^{(ML)}(S_i)$ for contextual splitting and $G_t^{(ML)}(S_i)$ for temporal splitting are

$$G_c^{(ML)}(S_i) = G_{output}(S_i), \tag{4}$$

$$G_t^{(ML)}(S_i) = G_{output}(S_i) + G_{trans}(S_i). \tag{5}$$

Eqs. (4) and (5) are calculated for each state, and the state with the maximum gain is selected.

3. MDL CRITERION

The MDL criterion[5] is one of the most popular information criteria and is used for the selection of the optimal model for stochastic models. Generally, when a set of models $\{\theta^{(i)}|i = 1, \ldots, I\}$ is given, the MDL criterion for model i is

$$L_i(\mathbf{x}) = -\log P(\mathbf{x}|\hat{\theta}^{(i)}) + \frac{\alpha_i}{2}\log N_T + \log I, \tag{6}$$

where $\mathbf{x} = \{x_1, \ldots, x_{N_T}\}$ is observation data, α_i is the number of free parameters, and $\hat{\theta}^{(i)}$ is the ML estimates of model i.

4. SSS ALGORITHM USING THE MDL CRITERION

Figure 2 shows the flow of MDL-SSS. The difference of MDL values for both contextual and temporal splitting is calculated for each state, and the state with the minimum difference value is selected as the split state. Splitting is finished when there is no state that can be split and reduce the criterion by splitting.

We define the criteria for contextual splitting and temporal splitting, $G_c^{(MDL)}$ and $G_t^{(MDL)}$, respectively, as the following:

$$G_c^{(MDL)}(S_i) = -G_c^{(ML)}(S_i) + C_c\frac{\alpha_c' - \alpha_c}{2}\log\Gamma(S), \tag{7}$$

$$\begin{aligned} G_t^{(MDL)}(S_i) &= -G_t^{(ML)}(S_i) + C_t\left\{\frac{\alpha_t'}{2}\log\Gamma'(S)\right. \\ &\quad \left. - \frac{\alpha_t}{2}\log\Gamma(S)\right\}. \end{aligned} \tag{8}$$

The first terms in the right-hand sides are the negative values of the expected gains in ML-SSS. C_c and C_t are scaling factors of second terms, which are not derived from the original definition of Eq. (6). We'll explain them later.

$\Gamma(S) = \sum_i \Gamma(S_i)$ represents the expected frequency of the number of samples for all states. $\Gamma'(S)$ is the value after temporal splitting. Eq. (8) compensates the total number of samples because segments that are shorter than the lengths of state sequences are discarded. α_c, α_c' are the number of parameters before and after contextual splitting, respectively. $\alpha_c = 2KM, \alpha_c' = 2K(M+1)$, when the order of features is K, the total number of states is M, and each state has one Gaussian distribution with a diagonal covariance matrix. For temporal splitting, we suppose that transition probabilities do not depend on both mean vectors and covariances of Gaussian mixtures. Each state has one Gaussian distribution

and one transition probability. Therefore, the number of parameters before and after temporal splitting are $\alpha_t = (2K+1)M$ and $\alpha'_t = (2K+1)(M+1)$, respectively.

The scaling factors, C_c and C_t, are not derived from the original MDL criterion. We experimentally found that it is difficult to stop splitting without the factors. This problem can be considered to be caused by the approximation of the likelihood values of temporal split states as we describe in Section 2. In [2], a scaling factor for contextual splitting was also introduced and experimentally found to be effective.

The MDL-SSS algorithm selects the state with the smallest $G_c^{(MDL)}$ or $G_t^{(MDL)}$, and stops splitting when $G_c^{(MDL)} > 0$ and $G_t^{(MDL)} > 0$ for all states.

5. EXPERIMENTS

5.1. Conditions

For the acoustic training set, we used dialog speech (5 hours in total) from the ATR travel arrangement task (TRA) database and read speech (25 hours) of phonetically balanced sentences (BLA). The same 407 speakers uttered both spontaneous and read speech. For testing, we used dialog speech from the TRA database uttered by a different set of 42 speakers. The sampling frequency was 16 kHz, the frame length was 20 msec, and the frame shift was 10 msec. 12-order MFCC, ΔMFCC, and Δ log power were used as feature parameters. The cepstrum mean subtraction was applied for each utterance. We used 26 kinds of phoneme models and one silence model with three states. One Gaussian distribution for each state was used during topology training. In Section 5.2, we used speaker independent models with one Gaussian distribution per state. These models could not always produce high performance. Therefore, after we obtained the topology, we increased the number of mixtures and re-estimated the parameters of HMMs. The final models were gender-dependent models with five Gaussian mixtures for each state. These models were used in experiments after Section 5.3. For the language training set, we used 7,195 one-side dialogues which included 1.6×10^6 words. Multi-class composite bigram models [6] were used as the language models. The full vocabulary size in the set was 27,398.

5.2. Comparison of models with one distribution per state

First, we investigated the performance by models with one Gaussian distribution for each state. In this section, the lexicon had only 5,100 words including words in evaluation data. These models did not have sufficient performance and our decoder could not obtain results by using the full lexicon. Figure 3 shows word accuracy rates by ML-SSS and MDL-SSS. MDL-SSS with $C_c = 2$ and $C_t = 20$ obtained almost the same performance as ML-SSS. For MDL-SSS, $C_c = 2$ and $C_t = 20$ performed best and for ML-SSS, $N_s = 2500$ and $N_p = 4$ showed the best performance.

5.3. Comparison of models with five mixtures per state

In this section, we used gender-dependent models with five Gaussian mixtures for each state. Figure 4 shows the word accuracy rates of these models. For MDL-SSS, $C_c = 2$ and $C_t = 20$ that were the same values as the previous section performed the best. For ML-SSS, $N_s = 1400$ and $N_p = 4$ performed the best. Therefore, for ML-SSS, N_s should be carefully adjusted according to the experiments to find the best model.

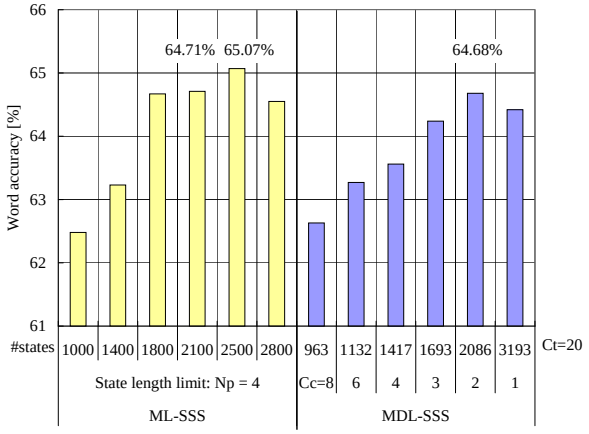

Fig. 3. Word accuracy for models with one Gaussian distribution per state.

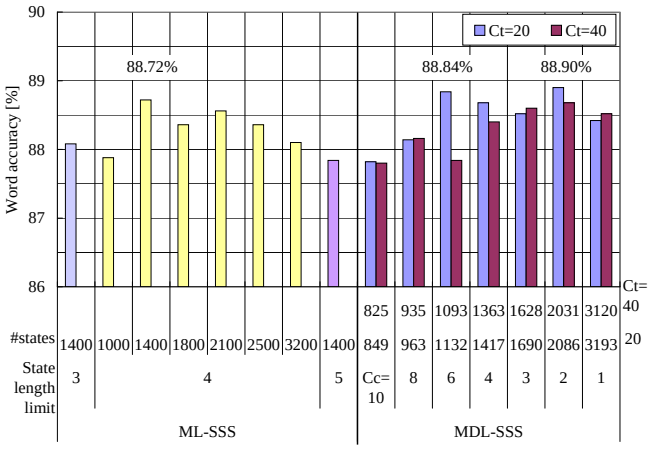

Fig. 4. Word accuracy for models with five Gaussian mixtures per state trained by using TRA and BLA.

We further compared the models of MDL-SSS and ML-SSS in detail. Figure 5 shows the maximum path length for each phoneme model extracted from both "ML-SSS (1400 states)" whose paths were set to a limit of four states, and "MDL-SSS ($C_c = 6$, $C_t = 20$, 1132 states)." All phoneme models by ML-SSS had the same maximum path length as the path limit number. On the other hand, each phoneme model by MDL-SSS had a different maximum path length. This suggests that more adequate path lengths are selected for each allophone by using MDL-SSS.

6. MDL-SSS USING DECISION TREE CLUSTERING

ML-SSS has one more problem in that it cannot deal with unseen contexts because of the data-driven clustering developed by P. A. Chou's algorithm[7]. In [8], phonetic decision tree (PDT) clustering is introduced into ML-SSS. They claim that the combination of decision tree clustering and data-driven clustering is best. We evaluated MDL-SSS using decision tree clustering.

6.1. Six different methods of context clustering

The contextual splitting in ML-SSS includes the two steps.

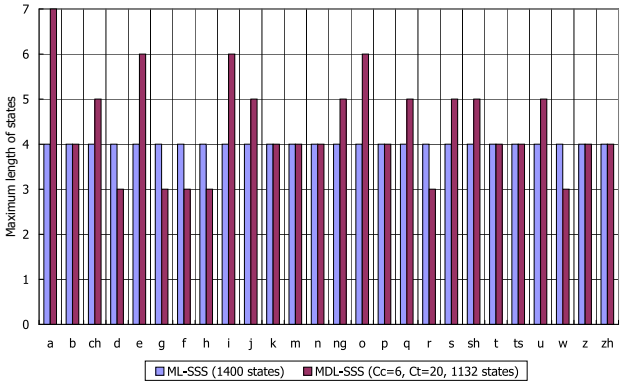

Fig. 5. The maximum length of paths for each phoneme.

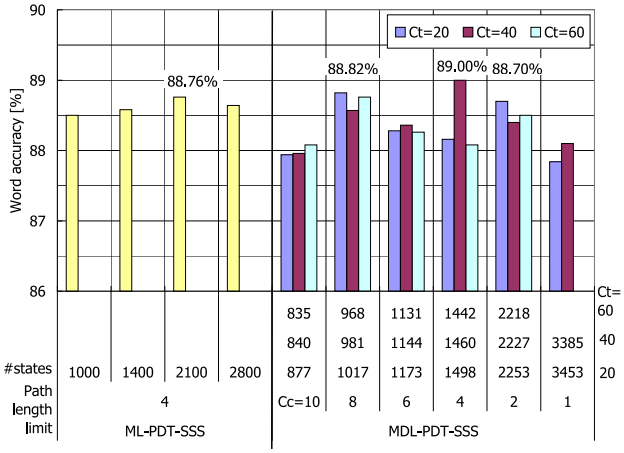

Fig. 6. Word accuracy by SSS using decision tree clustering.

1. **Generation of initial distributions:** two distributions are generated from original one and training data is classified.
2. **Iteration:** repeatedly two new distributions are recalculated and training data is classified.

In this paper, we evaluated the following six methods:

{**ML, MDL**}**-SSS:** Chou's algorithm is used repeatedly. The results are shown in Fig. 4.

{**ML, MDL**}**-PDT-SSS:** Decision tree clustering is used.

{**ML, MDL**}**-PDT+Chou-SSS:** First, decision tree clustering is used to create two initial distributions from one distribution. Second, Chou's algorithm is used repeatedly.

6.2. Evaluation of MDL-SSS with decision-tree clustering

The number of phonetic categories for questions was 47. The unseen contexts in the evaluation data were 1.1%. The experimental conditions were the same as Section 5.3. Figure 6 shows the performance by {ML, MDL}-PDT-SSS. It was difficult to set the best coefficients for C_c and C_t of MDL-PDT-SSS. Figure 7 shows the performance by {ML, MDL}-PDT+Chou-SSS. As with MDL-SSS, MDL-PDT+Chou-SSS could automatically obtain almost the same performance as ML-PDT+Chou-SSS. The best C_c, C_t of MDL-PDT+Chou-SSS were different from the best C_c, C_t of MDL-SSS.

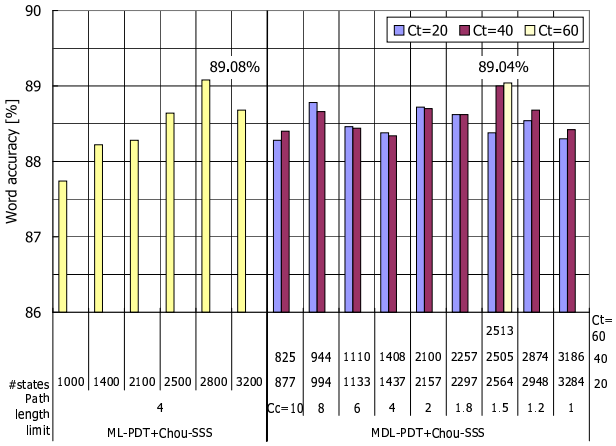

Fig. 7. Word accuracy by SSS using decision tree clustering and Chou algorithm.

7. CONCLUSION

We proposed the Successive State Splitting algorithm in conjunction with the MDL criterion. We introduced the MDL criterion to the ML-SSS algorithm in order to be able to select suitable models automatically. Experimental results show that the MDL criterion can stop both contextual and temporal state splitting by the SSS algorithm. Furthermore, we investigated two kinds of MDL-SSS combined with phonetic decision tree clustering. These methods were able to obtain a network topology automatically.

ACKNOWLEDGMENTS

This research was supported in part by the Telecommunications Advancement Organization of Japan.

8. REFERENCES

[1] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proc. of the ARPA Workshop on Human Language Technology*, pp. 307–312, 1994.

[2] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *The Journal of the Acoustical Society of Japan (E)*, vol. 21, no. 2, pp. 79–86, 2000.

[3] J. Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling," in *Proc. ICASSP'92*, vol. 1, pp. 573–576, 1992.

[4] M. Ostendorf and H. Singer, "HMM topology design using maximum likelihood successive state splitting," *Computer Speech and Language*, vol. 11, pp. 17–41, 1997.

[5] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. on IT*, vol. IT-30, no. 4, pp. 629–636, 1984.

[6] H. Yamamoto and Y. Sagisaka, "Multi-class composite n-gram based on connection direction," in *Proc. of ICASSP'99*, vol. 1, pp. 533–536, 1999.

[7] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 340–354, 1991.

[8] H. Singer and A. Nakamura, "Unified framework for acoustic topology modelling: ML-SSS and question-based decision trees," in *Proc. of EUROSPEECH'99*, vol. 3, pp. 1355–1358, 1999.