Construction of Phoneme Models

— Model Search of Hidden Markov Models —

Shiro Ikeda†

† Department of Mathematical Engineering and Information Physics,
Faculty of Engineering, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan,
Tel: +81-3-3812-2111 ext. 6883,
Fax: +81-3-5802-2913
E-mail: shiro@bcl.t.u-tokyo.ac.jp

Abstract: The author proposes an algorithm to define a structure of a HMM (hidden Markov model) [1]. HMMs are widely used in the speech recognition systems and at that time structures are usually determined according to the heuristic knowledge. In this article this problem is treated as so-called "model selection" problem in statistics. Two recognition experiments using this algorithm are shown. First, artificial data then, ATR speech database are used for the source. Through these experiments, the author shows that such model selection is effective.

1 Introduction

In the field of speech recognition, HMMs are widely used to represent the phones and words[2]. When we use the HMMs, we have to choose their structures at first. The structure of a HMM is represented by the number of the states and the definition of the transitions between them. In most of the conventional works using HMMs, the structures are chosen according to the heuristic knowledge.

The essence of this problem is how to choose the structure of a probability model. This problem is called "model selection" in statistics and some information criteria are proposed for this. One of the most popular criteria is AIC (Akaike's information criterion). We can execute model selection by selecting the model which minimizes the criterion. AIC can be used for the models whose parameters are estimated through maximum likelihood estimation, and can be used for HMMs whose parameters are estimated by Baum-Welch (B-W) algorithm[2].

When we execute model selection, we usually prepare a lot of models with different structures, and select the best one. But as we can see, HMM is a complicated probability model and there are enormous numbers of possible structures. Moreover, B-W algorithm is an iterating procedure and takes time. Instead of preparing models, the author proposes an algorithm to search the best structure by changing its structure successively. At this time, AIC is used to evaluate the models. AIC is applicable to compare the models which form a hierarchy, where the simple models are a sub-model of the complicated model.

In this algorithm, 1) A model of a simple structure is modified successively to a more complicated model by adding the states and the transitions between them. 2) At each point the parameters are estimated by B-W algorithm and evaluate the model by AIC. 3) If AIC was reduced, this algorithm continues to modify the model, otherwise quit. By these processes, we try to find the model which minimizes AIC.

In Section **4** , the author shows the results of two experiments. In the first one, hidden Markov probabilistic sources are used for the sources. And the task is to recognize the source to which each data belongs. The other experiment is 5 phones recognition with the ATR speech database .

Takami and Sagayama[3] proposed an Successive State Splitting Algorithm (SSS) to generate HM-Net (hidden Markov network). SSS is the algorithm to construct a network for whole phones at once. Though in this article, the purpose is to construct an HMM for each category separately. Further discussion will be given in Section **5** .

2 AIC

The information criterion AIC[4][5] is defined as the predictive error (Kullback-Leibler discrepancy) between the true distribution and the distribution of the model. Kullback-Leibler discrepancy is defined by

$$\begin{aligned} D(p,f) &= \int p(y)\log\frac{p(y)}{f(y|\theta)}dy \\ &= \int p(y)\log p(y)dy - \int p(y)\log f(y|\theta)dy \end{aligned} \tag{1}$$

where y is the data (in this article y is discrete), $p(y)$ is the true distribution, and $f(y|\theta)$ is the distribution of the model. This can be treated like the "distance" between two distributions, thus we want to select the model which minimize this value. Because the first term of the eq. (1) is independent to the model, we have to discuss only the

second term. Apparently the parameters which minimize the term are the maximum likelihood estimators θ^*. As the data set $Y = \{y\}$ is a sample from a distribution with distribution function $p(y)$, θ^* and $D(p, f)$ distribute around their true values θ_o and $D_o(p, f)$. Therefore, instead of using $D(p, f)$ for the criterion, we should use the estimate value of $D(p, f)$. The estimate value of $D(p, f)$ over the distribution of the training data is

$$E_p^Y[D(p, f)]. \tag{2}$$

Of course, it is impossible to know the true distribution p. By estimating this distribution from training data, and by doubling it, we have the definition of AIC by

$$\mathrm{AIC} = (-2)\sum_{i=1}^{n} \ln f(y_i|\theta^*) + 2m \tag{3}$$

where m indicates the number of the parameters.

AIC consists of two terms. One is the log likelihood of the data which shows the fitness of the model to the training data. The other term is the number of the parameters which shows the complexity of the model. We can denote the given data more precisely with more parameters. But, on the other hand this implies that the model would not be good for other data. Thus AIC is the criterion which define the "goodness" of the data by the balance of these two terms.

When we use AIC for HMMs, we have to note that m in AIC indicates the number of the mutually independent parameters. It means that every parameter has to satisfy the eq. (4).

$$\left.\frac{\partial f(y|\theta)}{\partial \theta_i}\right|_{\theta^*} = 0 \qquad for\ \forall\ i \tag{4}$$

Not all of the parameters of a HMM, which are estimated through B-W algorithm, are independent. But we can easily avoid this problem. For example, by replacing an a_{ij} which is not zero, with

$$a_{ij} = 1 - \sum_{j':j' \neq j} a_{ij'}$$

we can define AIC of HMM. For HMM whose output probability distributions are discrete, we can define AIC by

$$\mathrm{AIC} = (-2)\sum_{i=1}^{n} \log f(y_i|\theta^*) + 2(m - 2s) \tag{5}$$

where s indicates the number of the states.

3 The Proposed Algorithm

The algorithm is shown schematically in Figure 1. The structure of a HMM is represented by the number of the states and the definition of the transitions between them. In the algorithm, a model of simple structure is modified by adding the state and the transitions. First, the procedure of adding the states is shown, then that of adding the transitions is shown.

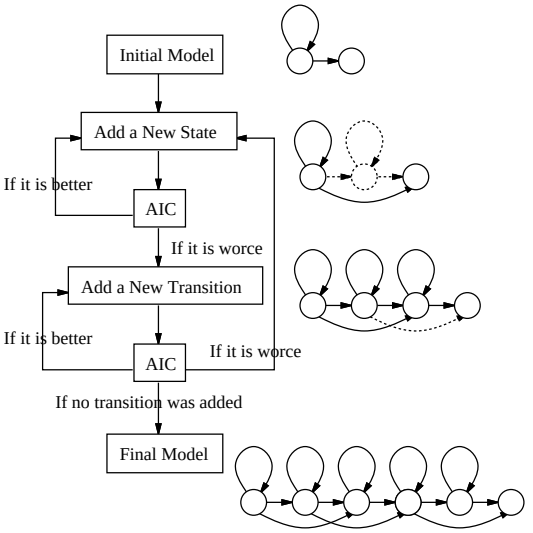

Figure 1: Flow of the algorithm

3.1 Adding New States

In this procedure, we increase the states one by one. To define a new state, first, we have to determine its position, then give initial values for the parameters of it and estimate the parameters by B-W algorithm. By iterating this, the number of the states are increased, and we quit increasing them according to AIC.

The position of the new state

If one of the states does not represent the data well, by dividing the state and giving more parameters, we can give more capability to the model. To see how "bad" the state is, the entropy of the output probability distribution of a state and the expect time to remain on the state are used. The formulations of them are as following

$$\text{Expected time to remain on the state } i \propto \sum_{t=1}^{T-1}\sum_{j=1}^{N} \alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1) \tag{6}$$

$$\text{Entropy of the state } i = \sum_{k}^{K} b_i(k)\log b_i(k) \tag{7}$$

The product of these two values is used to define the "badness" of the state, and the worst state is to be divided.

$$\text{The value of "badness"} = \sum_{t=1}^{T-1}\sum_{j=1}^{N} \alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1) \times \sum_{k} b_i(k)\log b_i(k). \tag{8}$$

Initial values

After dividing a state, we have to give initial values for the parameters. Then estimate the parameters by B-W algorithm. In the left model of Figure 2, the probability

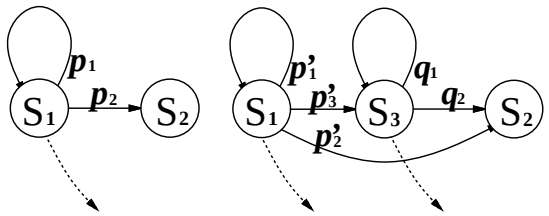

Figure 2: Define a new state

of staying n times at the State S_1 is

$$P_n = p_1^n p_2. \tag{9}$$

In the right model of Figure 2, the probability is

$$\begin{aligned} P'_n &= p'^n_1 p'^n_2 + p'^n_3 q_2^n \sum_{i=0}^{n-1} p'^i_1 q_1^{n-i-1} \\ &= \begin{cases} p'^n_1 p'_2 + q_2 p'_3 \dfrac{p'^n_1 - q_1^n}{p'_1 - q_1} & p'_1 \neq q_1 \\ \\ p'^n_1 p'_2 + n q_2 p'_3 p'^{n-1}_1 & p'_1 = q_1. \end{cases} \end{aligned} \tag{10}$$

If $p'_2 = p_2$, $q_1 = p_1$, $q_2 = p_2$, and $p'_1 + p'_3 = p_1$, P'_n of the eq. (10) is equal to P_n of the eq. (9). Therefore, if the output probability distribution of S_1 and S_3 are the same, the two models in Figure 2 are equivalent. This means that if we use these parameters for the initial values, the likelihood function of the model will be the same after dividing the state and estimating the parameters. In this procedure, it is clear that a sort of hierarchy is constructed among the models.

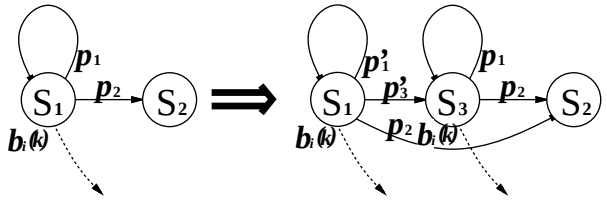

Figure 3: Initial values of the parameters

Exiting this Procedure

AIC which was given in Section **2** is used to evaluate the model, and if AIC is not reduced by adding a new state, we stop adding a state.

The procedure of increasing the states is as following.

1. Define the state to be divided.
2. Add a new state and set the initial values for the new parameters.
3. Estimate the parameters by B-W Algorithm.
4. Calculate AIC, and if the model is better than the last time, go to 1, otherwise quit.

3.2 Adding New Transitions between The States

After increasing the states, there is another problem that where to define the transitions. As we did on the states, we think about increasing the transitions one by one.

It is different from increasing the states that we cannot make the equivalent models when we put a new transition. This means that when we put a new transition, there are the risk of making the model worse. And there are no way except trying B-W algorithm to know if it made the model better.

Because it takes time to evaluate each possible transition by adding and estimating the parameters, we try to predict the increase of the likelihood function. For prediction, it seems better to use the increase of the one step of B-W algorithm, but it also costs enormous time because the number of possible transitions are large. Thus we use the inner product of the differences of one step of the B-W algorithm $(\hat{\theta} - \theta)$ and the derivative $\partial \log P(y|\theta)/\partial\theta$. This value can be calculated for all possible transitions by just one step of B-W algorithm. The detail of the calculation is shown in the following.

Increase of the transition probability

First of all, B-W algorithm for $\{a_{ij}\}$ is

$$\hat{a}_{ij} = \frac{a_{ij} \dfrac{\partial \log P(y|\theta)}{\partial a_{ij}}}{\displaystyle\sum_j a_{ij} \frac{\partial \log P(y|\theta)}{\partial a_{ij}}}. \tag{11}$$

By the eq. (11), it is clear that if a_{ij} is 0, a_{ij} does not change by B-W algorithm. Thus the structure of the HMM is not changed by B-W algorithm. When we try to put a new transition, first we set the value of it a little value δ and test $\hat{a}_{ij}$.

$$\hat{a}_{ij} = \frac{a_{ij} \dfrac{\partial \log P(y|\theta)}{\partial a_{ij}}}{\displaystyle\sum_{j:j\neq k} a_{ij} \frac{\partial \log P(y|\theta)}{\partial a_{ij}} + \delta \frac{\partial \log P(y|\theta)}{\partial a_{ik}}}, \ j \neq k \tag{12}$$

$$\hat{a}_{ik} = \frac{\delta \dfrac{\partial \log P(y|\theta)}{\partial a_{ik}}}{\displaystyle\sum_{j:j\neq k} a_{ij} \frac{\partial \log P(y|\theta)}{\partial a_{ij}} + \delta \frac{\partial \log P(y|\theta)}{\partial a_{ik}}} \tag{13}$$

The each of $\partial \log P(y|\theta)/\partial a_{ik}$ is finite value. If δ is very little value, the eqs. (12) and (13), the term of δ in their denominator can be neglected. Thus under this condition, the a_{ij} $(j \neq k)$ make almost no changes. With the following equation,

$$\left.\frac{\partial \log P(y|\theta)}{\partial a_{ij}}\right|_{\theta^*} = \left.\frac{\partial \log P(y|\theta)}{\partial a_{ij'}}\right|_{\theta^*}, \quad \text{for } \forall\ j' \ s.t.\ a_{ij'} \neq 0 \tag{14}$$

and $\sum_j a_{ij} = 1$, increase of a new transition a_{ik} is

$$\begin{aligned}\hat{a}_{ik} - a_{ik} &= \frac{\delta \dfrac{\partial \log P(y|\theta)}{\partial a_{ik}}}{\displaystyle\sum_{j:j\neq k} a_{ij}\frac{\partial \log P(y|\theta)}{\partial a_{ij}} + \delta\frac{\partial \log P(y|\theta)}{\partial a_{ik}}} - \delta \\ &\simeq \delta\left[\frac{\dfrac{\partial \log P(y|\theta)}{\partial a_{ik}}}{\dfrac{\partial \log P(y|\theta)}{\partial a_{ij}}} - 1\right], \quad j \neq k \wedge a_{ij} \neq 0. \end{aligned} \tag{15}$$

The $\{\pi_i\}$, $\{b_i(k)\}$ do not change their values if δ is very small. Therefore, we do not have to know the differences of the all parameters but of a_{ik} and the value can be estimated from the eq. (15). If $\hat{a}_{ik} - a_{ik}$ is positive, we think that it means that we should put the transition.

Increase of the likelihood function

On the other hand, what will the increase of the likelihood function be? In this algorithm, the inner product of $(\hat{\theta} - \theta)$ and $\partial \log P(y|\theta)/\partial\theta$ is used to estimate the increase. Taking it into account that a_{ij}, $(j \neq k)$, $\{b_i(k)\}$, and $\{\pi_i\}$ do not change their values, it is clear that the inner product is equal to the product of $(\hat{a}_{ik} - a_{ik})$ and $\partial \log P(y|\theta)/\partial a_{ij}$.

$\partial \log P(y|\theta)/\partial a_{ij}$ is easily obtained from B-W algorithm

$$\frac{\partial \log P(y|\theta)}{\partial a_{ij}} = \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_i(t) b_j(y_{t+1}) \beta_j(t+1)}{\displaystyle\sum_i \alpha_i(T)}. \tag{16}$$

Thus the increase of the log likelihood function is

$$\begin{aligned}\Delta_{ij} \log P(y|\theta) &\equiv (\hat{\theta} - \theta)\frac{\partial \log P(y|\theta)}{\partial\theta} \\ &\simeq \delta\left[\frac{\partial \log P(y|\theta)}{\partial a_{ik}} \Big/ \frac{\partial \log P(y|\theta)}{\partial a_{ij}} - 1\right] \\ &\quad \times \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_i(t) b_k(y_{t+1}) \beta_k(t+1)}{\displaystyle\sum_i \alpha_i(T)}. \end{aligned} \tag{17}$$

After calculating $\{\Delta_{ij} \log P(y|\theta)\}$ for all possible transitions, we decide to put the one which gives the largest value. In this procedure, only one step of B-W algorithm is enough for calculating all $\{\Delta_{ij} \log P(y|\theta)\}$.

Setting the probability of the new transition some small value, execute B-W algorithm and construct the new model. As in the last subsection, stop this procedure by evaluating each model by AIC and select the model which minimizes AIC.

4 Experiments and Results

4.1 Artificial Data

Before using acoustic data, we did an experiment with artificial data. The sources of the sequential data are generated by 5 HMMs. The task is to tell the model by which each datum was generated. The number of the states or the transitions of HMMs and also the output probability distributions are different. The source models are shown in Figure 4. There are 6 symbols and the probability distributions of the states are discrete.

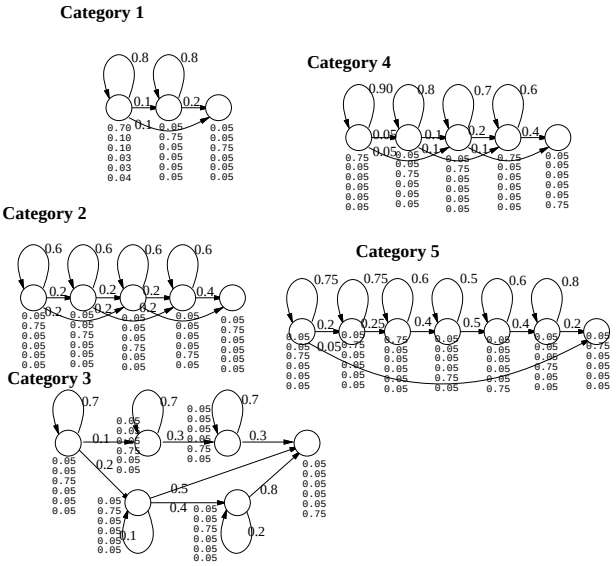

Figure 4: HMM for the probabilistic sources

The process of the experiment is as following.

- Let each model generate 2000 of sequential data. 1000 for training, and 1000 for recognition.
- Construct HMM for each category using the algorithm.
- Recognize the data according to the Bayes theory where the M_i which maximized $P(y|M_i)$ is the answer.

Result

The models which was constructed through the proposed algorithm are shown in Figure 5. Some of them reflect the structures of the source models but not all.

For comparison, the same recognition experiment with conventional method using the HMMs which have 3, 5, and 7 states are shown in the Table 1.

Figure 6 shows the log likelihood for Category 3. From this, the model constructed by the algorithm is good for both of the data for training and recognition.

Table 1: Recognition Accuracy

Sources	3 states	5 states	7 states	Algorithm
92.56%	84.92%	90.22%	90.68%	91.94%

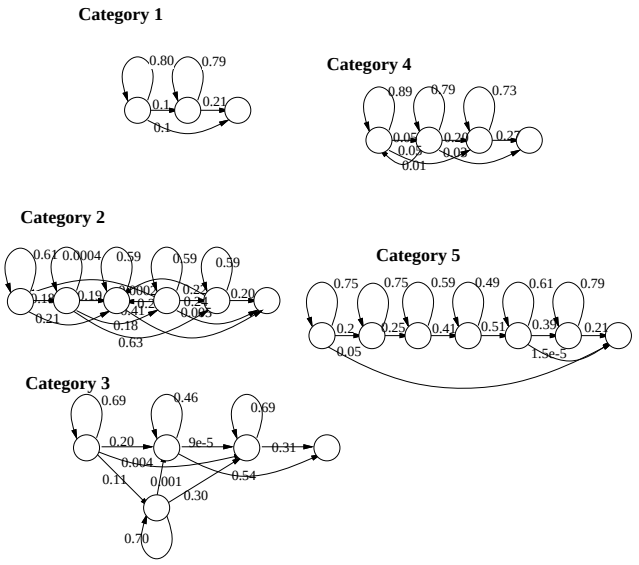

Figure 5: Results of the algorithm

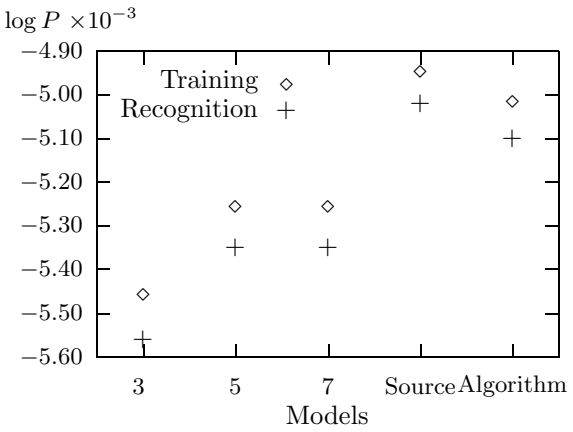

Figure 6: Log likelihood of each model

4.2 Acoustic Data

In this section, the experiment on phone recognition with this algorithm is shown.

Task Recognition of 5 phones (`/m/`, `/s/`, `/t/`, `/k/`, `/d/`) whose successive phone is `/e/`.

The Acoustic Data Speaker (MAU) in the Speech Database of ATR [6] and the phones are cut according to the labels [7].

Data for Vector Quantize 351 word of the task code B,NA,NB,SY,and F in the Speech Database of ATR.

Training data Odd numbers of 5240 words.

Data for recognition Even numbers of 5240 words.

Signal Processing Sampling 20kHz, 16bit, pre-emphasized with $1 - 0.97z^{-1}$, Hamming Window with the width of 20msec applied every 5msec.

Definitions of Symbol Execute VQ in [8] using vectors of 32 dimension which consists of logpow, mel-cep(15), Δ logpow, Δ mel-cep(15). And define 256 symbols.

Table 2: The HMMs constructed with the algorithm

phone(data)	m(72)	k(107)	t(66)	s(75)	d(28)
States	11	20	20	13	9
Transitions	32	71	64	40	27

Results are shown in Table 3. For comparison, the results with conventional approach such that the structure of HMMs are fixed. In the table, the number of their states and error rates for data of training and recognition are shown. In Figure 7, the model for `/d/` which was constructed through the algorithm is shown.

Table 3: Error rates
(The upper row are the results of the recognition data
The second row are the results of the training data)

3	5	10	15	20	Algorithm
10.3%	9.2%	8.0%	11.2%	14.4%	8.3%
(3.7%)	(2.0%)	(2.3%)	(1.4%)	(1.7%)	(2.0%)

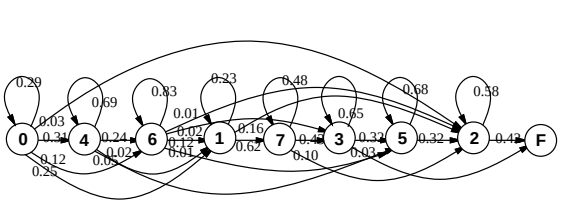

Figure 7: Constructed model for `/d/`

In Table 3, the models constructed through the algorithm performs like the fixed model which have 10 states. Though when we look at the log likelihood function, there are difference between them (Figure 8).

5 Discussion

5.1 The Results with The Algorithm

Figure 8 shows that the model with 15 states marked the best value for the training data and the model with the Algorithm is just the next. On the other hand, for the data of recognition, the model with the algorithm is the best.

In general, we can represent the training data more precisely with more parameters. But it is not the case with the data for recognition. By using AIC we can avoid the "over-learning" and construct good models. As is often the case with using some statistical approach, the number of training data must be large for this approach.

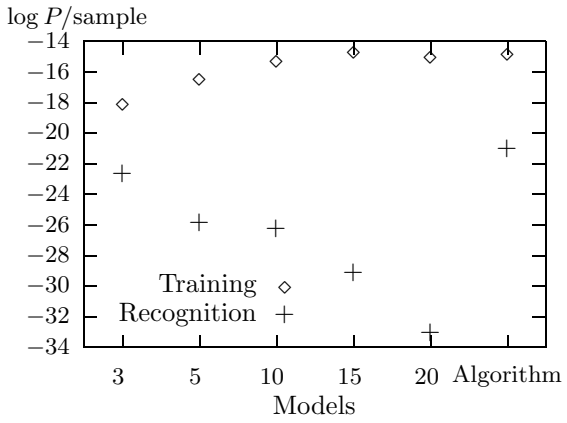

Figure 8: Log likelihood of the models for /m/

5.2 Likelihood and Recognition

With these experiments, it is shown that by searching the model which maximize the information criterion, we can have the model which represent the data better than the structures of the models are fixed. It seems that by using the constructed models, we can expect the recognition accuracy to be improved. Though what is shown in Table 3, the result of the constructed models is just similar to the models whose states are fixed to 10. Of course, the constructed models represents the data better than them as shown in Figure 8. The reason for this is that in these experiments, the model which gives the best value is the answer. Each model only has to output better value than other models. To represent the data precisely for such recognition experiments.

We have to note that this does not mean that model search is useless. What we have to consider when we construct the model is to represent the data precisely. Then what should we do to improve the recognition accuracy. Probably the things left for us to do is the classification of the data. If the data were categorized precisely, the recognition accuracy would be improved. This problem is equivalent to how to find the good phone units and is taken up in many speech recognition systems. For example, SPHINX system [2] used triphones (combinations of 48 basic phones) to make good phone units and defined 1076 phones. In SSS [3], they construct HM-Net from simple model which has only one state and treat the model selection and the classification at once. In this approach, by dividing the net into the contextual domain direction, they treat the problem.

The proposed algorithm is to construct a model for a given category. Therefore the classification must have done in advance. The problem of finding the good phone units is now under the consideration. As AIC is used in the model selection, the author are searching some good criterion for the classification. And then the recognition accuracy will be improved.

Besides the problem of the phone units, what the author is considering is to change the output probability functions of the HMMs. In the field of the automatic speech recognition, multi-dimensional Gauss distributions are widely used for the output probability distributions of the HMMs. With this we do not have to execute vector quantization. Now, the author is preparing for using this kind of HMMs for the experiments.

6 Conclusion

Through the experiments, the author shows that the model search approach is effective to construct the model which represent the data well. This means that when we implement this on some speech recognition system, the system can generate the efficient phone models by itself.

In the algorithm, the author used a theoretically derived criterion to determine the structure of the model instead of the heuristic knowledge. This will give some improvements in speech recognition systems.

References

[1] Shiro Ikeda, "Construction of Phone HMM using model search method", IEICE technical report SP93-26, IEICE, June 1993, (in Japanese).

[2] Kai-Fu Lee, *Automatic Speech Recognition — The Development of the SPHINX System*, Kluwer Academic Publishers, Norwell, Massachusetts, 1989.

[3] Jun-ichi Takami and Shigeki Sagayama, "Automatic Generation of the Hidden Markov Network by Successive State Splitting on Contextual Domain and Temporal Domain", IEICE technical report SP91-88, IEICE, Dec. 1991, (in Japanese).

[4] H. Akaike, "A new look at the statistical model identification", *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, 1974.

[5] Kei Takeuchi, "Distribution of information statistics and a criterion of model fitting", *Suri-Kagaku (Mathematical Sciences)*, pp. 12–18, Mar. 1976, (in Japanese).

[6] K Takeda, Y Sagisaka, S Katagiri, M Abe, and H Kuwabara, *Speech Database User's Manual*, ATR Interpreting Telephony Research Laboratories, 1986, (in Japanese).

[7] K Takeda, Y Sagisaka, S Katagiri, and H Kuwabara, *Manual Segmentation of Spectrogram for the Acoustic-Phonetic Transcriptions in Japanese Speech Database*, ATR Interpreting Telephony Research Laboratories, 1988, (in Japanese).

[8] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design", *IEEE Tr. Communications*, vol. COM-28, pp. 84–95, Jan 1980.