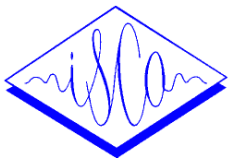

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

CONNECTED SPOKEN WORD RECOGNITION USING A MANY-STATE MARKOV MODEL

Tomio TAKARA, Naoto MATAYOSHI, and Kazuya HIGA

Department of Information Eng., College of Eng., University of the Ryukyus
1 Senbaru, Nishihara-cho, Okinawa-ken 903-01 Japan

Abstract

This paper is a report on an application of the Markov model to an automatic speech recognition system, in which a large number of states are adopted to model the transitional characteristics of speech more accurately. Unlike the traditional HMM, the feature vectors of this model are regarded to be the states of the Markov model. The transition-probability of the state is, in its initial condition, assumed to be represented by multidimensional normal density function of the feature vector. The many-state model is obtained by quantizing the feature vector (state) space and sampling the probability density function at each code vector. The resulting recognizer was tested and compared on a vocabulary of four-digit numerals using 3 dimensional feature vector sequences. The many-state model attained a recognition score of 98.2%, which was 1.6% higher than that of a five-state traditional HMM.

1 Introduction

The hidden Markov model (HMM) is applied to an automatic speech recognition system and the effectiveness of this model has been reported in many experimental studies [1]. In these HMMs, the symbol output probabilities are expressed with sufficient accuracy. However, the transition-probabilities of state are not expressed accurately: only a few parameters are used. These HMM models are, therefore, the models which set value on the symbol-output rather than the state-transition.

On the other hand, auditory feature of speech is thought to be expressed effectively by its transitional characteristics. Therefore, in order to construct a speech recognizer with high performance, it is better, from the view point of the Markov model, to set value on the state-transition rather than the symbol-output. A model such as this, called the continuous state model for the feature vector [2] has been previously proposed. The feature vector of this model is regarded to be the state of the Markov model.

However, the continuous state model has the problem that the computational load is large in the recognition mode: because the probability of this model is expressed by a continuous function; where, when a pattern is inputted, the likely-hood will be calculated every time.

In order to make the recognition process faster, we propose in this paper, a many-state Markov model for the feature vector. The many-state model is obtained by quantizing the feature vector space into many code vectors, then substituting code vectors into the probability density function of the continuous state model. This model can be adapted, by re-training, easily and effectively to be the model with more accurate probability distribution than the multidimensional normal density function of the continuous state model.

2 The Markov Model for the Feature Vector

A speech pattern X is expressed by a time sequence of feature vectors $\vec{x}_r$:

$$X = \vec{x}_1,\ \vec{x}_2,\ \dots,\ \vec{x}_r,\ \dots,\ \vec{x}_R, \tag{1}$$

$$\vec{x}_r = (x_{1,r},\ x_{2,r},\ \dots,\ x_{N,r}),\quad r = 1,\ 2,\ \dots,\ R. \tag{2}$$

where r, R and N are time (frame number), a total number of frames, and the dimension of the feature vectors, respectively.

We regard the sequence of the feature vectors to be a Markov process and each vector to represent a state and also an output symbol of the Markov model. When time changes from $k(q)$ to $l(q)$ and the state transits from $\vec{x}_{r(q)}$ to $\vec{x}_{s(q)}$, the probability $p(X)$ is expressed as follows:

$$p(X) = \sum_F p_1(\vec{x}_1) \times \prod_{q=1}^{Q} p_{l(q)|k(q)}(\vec{x}_{s(q)} \mid \vec{x}_{r(q)}) \times p_{k(q)}(\vec{x}_{r(q)}), \tag{3}$$

where $p_{l(q)|k(q)}(\vec{x}_{s(q)}|\vec{x}_{r(q)})$ is the transition probability of state, and $p_k(\vec{x}_r)$ is the symbol-output probability. $p_1(\vec{x}_1)$ is the probability density of the beginning point of the pattern. F is a function which represents a time sequence of the states, and is expressed by

$$F = c(1), c(2), \dots, c(q), \dots, c(Q), \tag{4}$$

$$c(q) = (r(q),\ s(q),\ k(q),\ l(q)), \tag{5}$$

where Q is the number of the state-transitions and the time changes from the beginning to the end point of the speech pattern X.

We assume $p_{l|k}(\vec{x}_s \mid \vec{x}_r)$ and $p_k(\vec{x}_r)$ are multidimensional normal density functions. Eq.(3) can be rewritten to be

$$p(X) = \sum_F p_1(\vec{x}_1) \times \prod_{q=1}^{Q} p_{k(q),l(q)}([\vec{x}_{r(q)},\ \vec{x}_{s(q)}]), \tag{6}$$

where $[\ \vec{x}_r\ ,\ \vec{x}_s\ ]$ represents a vector made by catenating the row vectors $\vec{x}_r$ and $\vec{x}_s$. $p_{k,l}([\ \vec{x}_r\ ,\ \vec{x}_s\ ])$ is also a multidimensional normal density function.

On the assumption that $p(X)$ of Eq.(6) is approximated by a maximum probability of a sequence which makes the largest contribution of F to $p(X)$, we have

$$p(X) \approx p(X_F) \tag{7}$$

$$= \max_F \left[p_1(\vec{x}_1) \prod_{q=1}^{Q} p_{k(q),l(q)}([\vec{x}_{r(q)}, \vec{x}_{s(q)}]) \right]. \tag{8}$$

Accordingly, if we take logarithm of Eq.(8) and give it the minus sign, we have

$$-\ln\ p(X_F) = d(1,1)$$

$$+ \min_F \sum_{q=1}^{Q} d(\ r(q), s(q), k(q), l(q)\ ), \tag{9}$$

where

$$d(r,s,k,l) = -\ln\ p_{k,l}(\ [\vec{x}_r,\ \vec{x}_s]\ ) \tag{10}$$

$$= -\ \ln\ p_{k,l}(\ \vec{y}_{r,s}\ ) \tag{11}$$

$$= 2^{-1}[\ (\ \vec{y}_{r,s} - \vec{v}_{k,l}\ )V_{k,l}^{-1}(\ \vec{y}_{r,s} - \vec{v}_{k,l}\ )^{\mathrm{T}} + \ln(2\pi)^{2N}|V_{k,l}\ |\ ], \tag{12}$$

$$\vec{y}_{r,s} = [\ \vec{x}_r, \vec{x}_s\ ], \tag{13}$$

$$d(1,1) = -\ln\ p_1(\vec{x}_1) \tag{14}$$

$$= 2^{-1}[(\ \vec{x}_1 - \vec{\xi}_1\ )V_1^{-1}(\ \vec{x}_1 - \vec{\xi}_1\ )^{\mathrm{T}} + \ln(2\pi)^{N}|V_1\ |\ ]. \tag{15}$$

In Eq.(12), $\vec{v}_{k,l}$ and $V_{k,l}$ are the mean vector and the covariance matrix of $\vec{y}_{k,l}$, respectively, and in Eq.(15), $\vec{\xi}_1$ and V_1 are the mean vector and the covariance matrix of $\vec{x}_1$, respectively. Eq.(9) can be efficiently evaluated by using the DP-matching algorithm [2].

The reference patterns of the current model for each word class involve sequences of the mean vectors and covariance matrices. We describe first, the context-independent training method [2] and next, the context-dependent method [3] for reference pattern generation of words.

First, in the context-independent reference pattern generation, one of the feature vector sequences for each word class is memorized in the system. We call these sequences: core patterns, which are used for arranging frames of inputted patterns. Next, another word pattern involved in the same word class is inputted and DP-matched to the core pattern. For the DP-matching, we adopt the slope constraint DP method [2],[3], in which the path which minimizes the time-normalized distance between the core pattern and the inputted pattern is selected. Along this path, the covariance matrices and the mean vectors at each frame of the core pattern are renewed using corresponding frames of the inputted pattern [2]. The covariance matrix $V_{k,l}$ is estimated as V_k; considering that covariance matrices with the same k are identical. One by one, all other training words of the same word class are inputted and the same procedure is repeated. The sequence of the covariance matrices and the mean vectors is gradually estimated. For each word class of a vocabulary, the above procedure is taken, and the reference patterns of the word classes are made.

In the context-dependent method of reference pattern generation, first, the context-independent reference patterns of word class, described above, are generated using isolated words for training. Next, the core pattern for the context-dependent method is made by catenating the mean vectors of the context-independent reference patterns according to a vocabulary of connected words. Taking a similar procedure to the context-independent method, DP-matching is done, and covariance matrices and mean vectors are renewed. The covariance matrices and the mean vectors of the context-independent method are used for initial values of those of the context-dependent method. Changing the combination of words of the core pattern, according to the vocabulary, the above procedure is repeated for all connected words. The sequences of covariance matrices and mean vectors are gradually estimated for each word class.

In order to apply the proposed stochastic model to the connected word recognition, we adopt the automaton-controlled One Pass DP method [5], which is a suitable algorithm for real time processing of speech recognition. Vocabulary of the connected words is controlled by a finite state automaton which gives a dictionary of the vocabulary. A category which minimizes the value of Eq. (9) is selected among categories of the connected word, and is determined as the category of the inputted speech pattern.

3 The Many-State Model

In this section, we propose the many-state Markov model for the feature vector by revising the model described in section 2 using vector-quantized parameter, and we describe the retraining method which makes the many-state model more effective.

3.1 Joint-Probability Matrix

First, using a clustering algorithm, the feature vector space is divided into many sub-spaces each of which is represented by the cluster center. Then, a code-book is made by adopting these cluster centers for code vectors. In this study, we use FMS [6] for the feature vector and the clustering algorithm in which the FMS-space is repeatedly divided into two sub-spaces, obtaining cluster centers which minimize the estimation error at each sub-space.

Construction of the many-state Markov model, that is to say, the generation of a reference pattern of the model is performed as follows: we assume here that the probability density functions represented by the covariance matrices and the mean vectors have already been obtained by the context-dependent reference pattern generation method described in section 2. Using these probability density functions, at each frame k for each word class, the joint-probability $p_{i,j}^{(k)}$ between all code vectors are computed using the following expression:

$$p_{i,j}^{(k)} = (2\pi)^{-N}|V_k|^{-1/2}$$

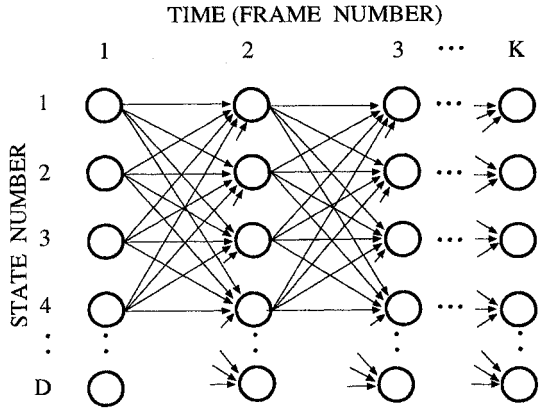

Fig.1. THE MANY-STATE MARKOV MODEL

$$\times \exp\left[\ 2^{-1}(\vec{y}_{i,j} - \vec{v}_{k,k+1})V_k^{-1}(\vec{y}_{i,j} - \vec{v}_{k,k+1})^{\mathrm{T}}\ \right], \quad (16)$$

$$\vec{y}_{i,j} = [\vec{x}_i, \vec{x}_j], \quad (17)$$

where $\vec{x}_i$ and $\vec{x}_j$ are the i-th and the j-th code vectors, respectively. And where $\vec{v}_{k,k+1}$ and V_k are a mean vector and a covariance matrix of Eq. (12), respectively. $\vec{y}_{i,j}$ represents the vector made by catenating two vectors which correspond to a source code i and a destination code j. The $D \times D$ dimensional joint-probability matrix is obtained by computing the Eq. (16) for all combination of codes.

The many-state Markov model is shown in Fig.1, in which the circle represents the state or the code at each time (frame). The arrow shows the direction of the transition. The probabilities corresponding to the arrows are represented by a joint-probability matrix for each frame.

If we adopt such matrices as they are, the system will need very large memory. Therefore, we compress the matrices as follows: first, the column factors are summed up at each row [8], resulting in values which represent probabilities for each code. Next the d-codes with the largest probabilities are selected and memorized as a table which we call an order vector. The factor numbers of the order vector represent the codes, and the values of the factor are the orders. The factors of the joint-probability matrix are re-arranged to make the orders be the factor numbers of a new matrix. Factors other than the d factors of the matrix are eliminated and regarded to have a small constant value b_k, which is only memorized in the system (where k is the frame number). The factors smaller than b_k are also changed to b_k in the compressed matrix. The above procedure is taken for all matrix of the frame. The factors of the compressed matrix are read via the order vector.

In the recognition (matching) mode, the joint-probability $p_{i,j}^{(k)}$ and b_k are used as the distances after the transformation where they are taken logarithm and given a minus sign.

3.2 Re-Training

First, similar to the context-dependent reference pattern generation, a core pattern is made by catenating the many-state model of words according to the vocabulary. Next, the core pattern is DP-matched to a connected word pattern for training represented by the code. As a local distance of the DP, we adopt a value of factor of the joint-probability matrix which corresponds to the codes of two time points. The factor is taken logarithm and given a minus sign. Along the optimal path of the DP, the joint-probability matrix is renewed. The m-th renewal of the (i,j)-factor $p_{i,j}^{(k)}(m)$ of a joint-probability matrix of the k-th frame is done as follows, using the former value $p_{i,j}^{(k)}(m-1)$:

[For the factor corresponding to the transition]

$$p_{i,j}^{(k)}(m) = A_k \cdot p_{i,j}^{(k)}(m-1) + B_k. \quad (18)$$

However, if the order of probability is lower than d, then

$$b_k(m) = S^{-1}[S \cdot A_k \cdot b_k(m-1) + B_k]. \quad (19)$$

[For the other factors]

$$p_{i,j}^{(k)}(m) = A_k \cdot p_{i,j}^{(k)}(m-1) \quad (20)$$

where

$$A_k = \frac{M_k(m-1)}{M_k(m-1)+1}, \quad (21)$$

$$B_k = \frac{1}{M_k(m-1)+1}, \quad (22)$$

$$S = D \cdot D - d \cdot d, \quad (23)$$

$$M_k(m) = M_k(m-1) + 1 \quad (24)$$

and $M_k(m-1)$ is the number of training data used before the m-th renewal.

3.3 Recognition Process

In the recognition mode, the automaton controlled One Pass DP is used similarly to the continuous state model described in section 2. The local distance of the many-state model is obtained directly by searching the joint-probability matrix via the order vector, whereas, the local distance of the continuous model is obtained by calculation. First, the order vector is checked to see if two codes of the current time points (frames) exist in the top d. If they are found, the factor of matrix at the row and the column corresponding to the two orders is determined to be the local distance. Otherwise, $-\ln b_k$ is adopted.

4 Recognition Experiment

The speech sampling rate is 10kHz, and overlapping sections of 25.6ms of speech weighted by the Blackman window are analyzed every 10ms to give the FFT power spectra. The power spectra are transformed to the FMSs [6], which are the Fourier transforms of Mel Sone spectra whose frequency-axes are warped to be the mel scale and magnitude-axes are warped to be the sone scale. Three dimensional vectors, whose components are second to fourth components of the FMS, are used as the feature vectors.

The speech data in the recognition test are English numeral words of TIDIGITS [9], from which we use, as isolated words, eleven single-digit numerals "one" to "nine", "zero" and "oh", uttered twice by 16 American males. As the connected words for training, we use 11 four-digit numerals uttered once by 20 males including the above 16 males. As the data for recognition test, we use four-digit numerals

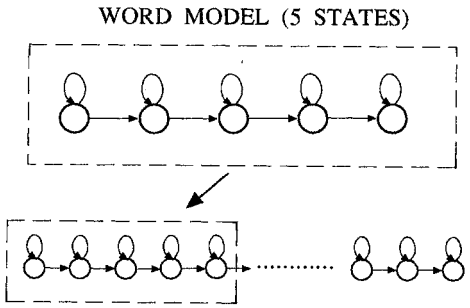

Fig.2 HMM USED FOR COMPARISON

TABLE 1. RESULT OF THE RECOGNITION EXPERIMENT(%)

		MODEL		
		MSMM+ RE-TRAINING	MSMM	HMM
CODE BOOK	A	98.5	97.5	97.0
	B	97.0	97.0	97.0
	C	99.0	97.5	94.9
AVERAGE		98.2	97.3	96.6

MSMM : MANY-STATE MARKOV MODEL
HMM : HIDDEN MARKOV MODEL

uttered once by another 18 males. Six open tests are done in the recognition experiment dividing 18 speakers into six groups of three members with a vocabulary size 33.

First, we generate the context-independent reference patterns of the continuous state model using the single-digit words. Next, using these patterns, we obtain the context-independent reference patterns of words using the four-digit numerals for training. And then, the many-state models are obtained by sampling the context-dependent model at each point of the code vectors. The code book is made from all feature vector obtained by FMS analysis of the four-digit numerals of training. The code book size 64 is adopted [8].

In order to evaluate the proposed model, we perform another recognition test using a typical HMM, which is shown in Fig.2. The model is 20 states discrete HMM, constructed by catenating four five-state word models. To achieve the same experimental condition as the proposed model, the feature vector is set to be three dimensional FMS, and the code book size is 64.

The results of the recognition tests are shown in TABLE 1, where three code books were made and used for each test. It is shown that the many-state model (MSMM) has a higher performance than the HMM. Furthermore, the MSMM +re-training's performance is higher than that of the MSMM, indicating the effectiveness of the re-training method.

In the Markov models for the feature vector (the continuous state model and the many-state model), the probability distributions are renewed only on the optimum path of the DP as described in section 2 and subsection 3.2. This corresponds to the situation in which only one path is used on the trellis of the HMM. Therefore, the processing time to generate the reference patterns of the proposed model is thought to be much shorter than that of the HMM. In these experiments, the processing times of the reference pattern and model generation were measured. It was found that the time of the many-state model is 1/50 of that of the HMM. The processing times of the recognition mode are almost the same in both models.

5 Conclusion

In order to make the recognition process faster in the continuous state Markov model for the feature vector, we propose the many-state Markov model for the feature vector. The many-state model is obtained by vector-quantizing the feature vector space and sampling the probability function of the continuous state model at each point of the code vectors. We also propose the re-training method for the many-state model. In the recognition experiment, we evaluated the performance of the proposed model compared to that of the typical HMM. The proposed model was shown to have a higher recognition score than that of the HMM and the processing speed for the reference pattern generation was 50 times faster than that of the HMM. Future plans are to study an optimal compression method for the joint-probability matrix and decision method for a optimal value of b_k.

References

[1] M. Okochi: "Speech Recognition Based on Hidden Markov Models", (in Japanese) The Journal of the Acoustical Society of Japan, 42, 12, pp.936-941 (Dec. 1986).

[2] T. Takara and T. Yakabu: " Connected Spoken Word Recognition Using the Markov Model for the Feature Vector ", IECE Trans., E74, 7, pp. 1788-1796 (Jul. 1991).

[3] T. Takara: "Connected Spoken word Recognition for English Using the Markov Model for the Feature Vector", (in Japanese) Conference of the Acoustical Society of Japan, 3-7-4, pp. 169-170 (Oct. 1992).

[4] H. Sakoe and S. Chiba: "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. Acoust., Speech Signal Process., ASSP-26 , 1, pp. 43-49 (Feb. 1978).

[5] S. Nakagawa: "Speech Recognition Using Probability Model", (in Japanese) pp. 18-26, Corona Co., Tokyo (1988).

[6] T. Takara and S. Imai: "Isolated Word Recognition Using DP-Matching and Maharanobis' Distance", (in Japanese) Trans. IECE Japan, J66-A, 1, pp. 64-70 (Jan. 1983).

[7] pp.26-28 in [5].

[8] T. Takara and S. Urasaki: "Connected Spoken Word Recognition Using the Discrete-State Markov Model for the Feature Vector", (in Japanese) Technical Report of IECE, SP93-52, pp.25-32 (Aug. 1993).

[9] NIST: "TIDIGITS CD-ROM Set", NIST (Feb. 1991)