Multi-group Mixture Weight HMM

Li Ming, Yu Tiecheng
Speech Processing Lab, Institute of Acoustics, Chinese Academy of Sciences
P.O.Box 2712, Beijing 100080, P.R.CHINA
Email: {lm,tcyu}@speech1.ioa.ac.cn

ABSTRACT

This paper presents a new modeling method of the continuous density Hidden Markov Model. As we know, speech signal is characterized by a hidden state sequence and each state is described by the mixture of weighted Gaussian density functions. Usually if we want to describe speech signal more precisely, we need to use more Gaussian functions for each state. But it will increase the computation significantly. On the other hand, the weight of each Gaussian component is the statistical average of Gaussian component probabilities for the whole training data. So it just can depict the average characteristics of speech signal. For some speech signal these weights are not proper in fact. Therefore, we propose Multi-group Mixture Weight HMM to solve this problem. In this kind of HMM, each state has several groups of mixture weight for the Gaussian components and it only needs very little additional computation. In our experiments, it achieved 12% reduction for errors.

1. INTRODUCTION

When continuous density HMM was applied to speech recognition[1][2], it achieved great success in recent years[3][4]. An HMM can be completely characterized by a matrix of state transition probabilities, observation densities, and initial state probabilities. Some study has found the observation densities are most important for HMMs. Most improvement of HMM is made on this respect. In continuous density HMM, states are characterized by the mixture of weighted Gaussian density functions. Generally speaking, if we want to improve the performance of recognizer, a straight forward way is to train HMM with more Gaussian components. Because logarithmic and exponential computation are very time-consuming, so it will take much more time for training and recognition than the HMMs with fewer Gaussian components. Thus we try to find a way that we can improve the performance and retain the number of Gaussian components at the same time. Multi-group Mixture Weight HMM is such a method. By this method we reduce errors with very little additional computation.

Another more important reason which motivated us to propose such a method is based on the following consideration. Observing the re-estimate equations[1][2], it can be found that the weight of each component is the statistical average of the component probabilities. So these weights can be used to describe the average characteristics of the corresponding state. Usually some components are distributed high weights while some are distributed low weights. Accordingly, the characteristics of these states is characterized mainly by these components which have high weights. Whereas, the characteristics of some speech is closer to that of those components which have low weights. Therefore, these speech is modeled improperly. In our method, each state has several groups of component weights. So it can meet different cases.

In section 2 we will give details about our new method. In section 3, we present the experiment results. The summary is given in section 4.

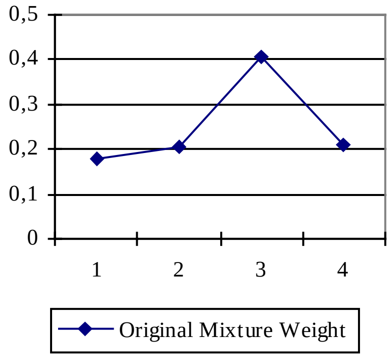

Fig. 1. The Distribution of the original mixture weight of four components

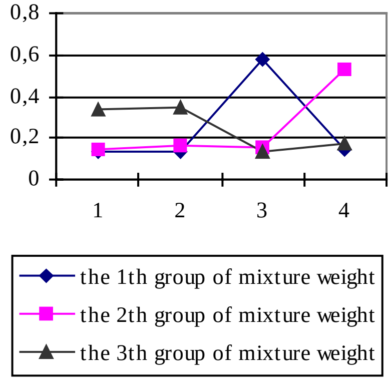

Fig. 2 The Distribution of three groups of mixture weight, which are derived from the above original mixture weight

2. MODELING METHOD OF MULTI-GROUP MIXTURE WEIGHT HMM

Our method can be briefly described as follows. Because of the importance of initial parameters for continuous density HMM, we first should get proper initial parameters of the component weights. At initial stage, for each frame in an utterance we calculate a vector called mixture weight vector and store it in the vector pool of the corresponding state. Then we classify all mixture vectors in the vector pool of a state into several clusters. The center vector of each cluster corresponds to a group of initial mixture weight. After initialization we can train continuous density HMM by EM algorithm[5][6]. For recognition, we can select an appropriate group of mixture weight to calculate the output probabilities.

Fig.1 shows an original mixture weight and Fig.2 shows three groups of mixture weight which are derived from the original mixture weight. In Fig.1 it can be seen that the highest weight is given to the third component in the original mixture weight. While in Fig.2, a group of mixture weight is similar with the original mixture weights, while the other two groups distribute the highest weights to the other components instead of the third component. Therefore, the second group and the third group can describe the speech more accurately whose characteristics is closer to that of the first, second or fourth component.

2.1 Retrieval of Initial Multi-group Mixture Weight

Because the initial parameters are essential to the quality of the final continuous density HMMs[1], so it is necessary to get proper initial parameters of multi-group mixture weights. For convenience, we introduce Equ.(1), a widely used equation in continuous density HMM[2]. In (1), $\xi_t(j,k)$ is the probability of taking all possible transition to state *j* and have the *k*th mixture component at time *t*, given the model λ and observation *X*.

$$\xi_t(j,k) = f\left(s_t = j, k_t = k \mid X, \lambda\right) \qquad 1 < t \le T \qquad (1)$$

For the observation vector X_t at time *t* and given the model λ, we define the following variables,

$$\eta_t(j) = \sum_{k=1}^{M} \xi_t(j,k) \qquad (2)$$

$$F_j(X_t) = \left\{ \frac{\xi_t(j,1)}{\eta_t(j)}, \frac{\xi_t(j,2)}{\eta_t(j)}, \cdots, \frac{\xi_t(j,M)}{\eta_t(j)} \right\} \qquad (3)$$

where, *M* is the number of mixture components, $F_j(X_t)$ is a M-dimension vector, called mixture weight vector.

In order to obtain the initial parameters of continuous density HMM, first we have to have some speech data which have been labeled by hand or by forced *Viterbi* decoding[7]. Then we calculate the mixture weight vectors of all labeled speech data by Equ. (3). After that, we put all mixture weight vectors $F_j(X_t)$ for the state *j* of the model λ together and classify them into several clusters by the classical clustering algorithm[8] . The center vector of each cluster represents a group of initial mixture weights.

But when observing the initial weights, we can find that some weights are so high nearly to 1 while some weights are almost zero. In such a case, these states will be characterized by only one Gaussian component actually, which would degrade the performance of the continuous density HMM. In order to relieve from such a problem, we smooth the mixture weights as below.

For a group of mixture weight $F = \{c_1, c_2, \cdots, c_M\}$, which satisfies $\sum_{i=1}^{M} c_i = 1$ we could smooth it by Equ. (4),

$$c_i^{'} = c_i * \theta + (1 - c_i) * (1 - \theta) \qquad 1 \le i \le \text{M} \qquad (4)$$

Where,

$$\theta = \frac{\varepsilon * (M-1)}{1 + \varepsilon * (M-2)} \qquad \varepsilon = 0.5 \sim 0.7 \qquad (5)$$

After such a transformation, $c_i^{'}$ $1 \le i \le M$ don't satisfy $\sum_{i=1}^{M} c_i^{'} = 1$ any more. So we should normalize them by their sum.

Let,

$$C^{'} = \sum_i c_i^{'} \qquad (6)$$

$$c_i^{''} = c_i^{'} / C^{'} \qquad 1 \le i \le M \qquad (7)$$

Then, the group of finally initial mixture weight will be $F = \{c_1^{''}, c_2^{''}, \cdots, c_M^{''}\}$.

2.2 Training of Multi-group Mixture Weight HMM

Given the observation vector X_t, we assume X_t is assigned to the state *j* of the model λ while *Viterbi* decoding[7]. We calculate the output probability and the mixture weight vector of X_t for each group of mixture weight respectively. Then we note down the mixture weight vector which gives the maximum output probability and the index of the mixture weight group, the state and the model which the vector belongs to.

After each iteration, we add up all mixture weight vectors which belong to the *p*th group, state *j* and model λ and divide the sum by the number of vectors. Then we get the *p*th group of mixture weight of state *j*, model λ for the current iteration. The re-estimate formula for the mixture weight can be written as (8). After a few iterations using EM algorithm[5][6], good model parameters will be obtained.

$$\hat{F}_{p,j,\lambda} = \frac{1}{T} \sum_{t=1}^{T} F_{p,j,\lambda}(t) \qquad (8)$$

Where, $F_{p,j,\lambda}(t)$ is the mixture weight vector at time *t* which is assigned to the *p*th group of mixture weight, state *j* and model λ after *Viterbi* decoding.

2.3 Recognition With Multi-group Mixture Weight

There is no much difference between the traditional HMM and multi-group mixture weight HMM for recognition. When calculating the output probability of every observation, we can just choose the group of mixture of mixture weight which outputs the maximum probability.

$$b_j(o_t) = \max_{p=1}^{P} \sum_{k=1}^{M} c_{p,k} N(o_t,\mu,\Sigma) \qquad (9)$$

Where, $b_j(o_t)$ is the output probability for observation o_t at state j, $c_{p,k}$ is the kth component weight of the pth group, $N(.)$ is the Gaussian density function.

When calculating the output probabilities, we can calculate the probabilities of the Gaussian density functions first, then multiple them with each group of mixture weight respectively. Thus we only need several multiplication for each state in addition. Compared with the exponential operation, this additional computation can be ignored.

3. EXPERIMENT RESULT

Our experiment is made on a digital string recognition system. The training data include 40 persons' speech data and the test data include 6 persons'. Each person has 50 utterances of digital strings.

In order to compare with the traditional continuous density HMM, we also made experiments with the traditional HMM which only has one group of mixture weight for each state. In our Multi-group Mixture Weight HMM, we use three groups of mixture weight for every state. Table1 shows the results of the two different modeling methods.

	Traditional HMM	MGMW HMM
Digital correct	97.68%	98.00%
Delete error	0.97%	0.90%
Substitute error	1.35%	1.09%
Insert error	0.00%	0.06%

Table1 The experiment results of the traditional HMM and Multi-group Mixture Weight HMM.

Fig. 3 shows we get about 12% error reduction compared with the traditional continuous HMM when using three groups of mixture weight, which indicates the effectiveness of Multi-group Mixture Weight HMM.

4. SUMMARY

In traditional continuous HMM, each state is characterized by the mixture of a few weighted Gaussian density functions. But the weight coefficients are fixed and there is only one group of them for every state. So it can not describe some speech signal very precisely because speech signal may have various characteristics even for one state. Multi-group mixture weight HMM solves this problem, which has several groups of mixture weight to match the different characteristics. The experiment shows that errors are declined about 12% when this method is applied.

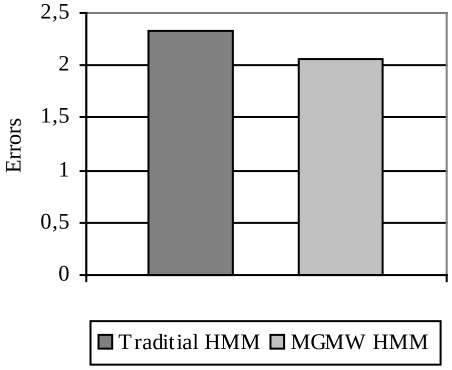

Fig3. The performance comparison of the traditional HMM and Multi-group Mixture Weight HMM

5. REFERENCES

[1] Rabiner L.R., Biing-Hwang Juang, "Fundamentals of Speech Recognition", Published by PTR Prentice-Hall Inc., 1993.

[2] Huang X.D., Ariki Y., Jack M.A., "Hidden Markov Models For Speech Recognition", Edinburgh University Press, 1990.

[3] X.D. Huang. F.Alleva. H.-W. Hon. M.-Y Hwang.K.-F "The SPHINX-II Speech Recognition System":An Overview." Computer Speech and Language Vol.2.pp 137-148. Feb. 1993

[4] L.R. Bahl. et al., "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task.", ICASSP'95 Vol.1 pp41-44. Detroit Michigan. U.S.A. May 1995

[5] Baum L.E., Petrie T., Soules G., Weiss N., "A maximum technique occurring in the statistical analysis of probabilistic functions of Markov chains", Ann. Math. Stat., Vol. 41, pp.164-171, 1970.

[6] Baum L.E., "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes", Inequalities, Vol. 3, pp.1-8, 1972.

[7] Viterbi A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE *Trans.* on Information Theory, IT-13(2), pp.260-269, April 1967.

[8] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizater Design", IEEE Tans, COM, Vol. COM-28, No.1, January, pp. 84-95, 1980