REVIEW OF MULTI-GROUP MIXTURE WEIGHT HMM

Peng Peng

Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762 USA
email: peng@isip.mstate.edu

ABSTRACT

Continuous density Hidden Markov Model method has been used successfully as the basic modeling technique in Automatic Speech Recognition. To improve the precision of our description of speech signal, we can use more Gaussian mixtures for each state. But it will increase the computation significantly. On the other hand, since the weight of each Gaussian component are not always the statistical average of Gaussian component probabilities, adjusting the weight should be another way which can affect the final error rates of speech recognition. This paper will focus on this idea and give out theoretical steps and practical operations which lead to an improvement in the final speech recognition result. The authors announce that they achieved 12% error reduction compared with the traditional continuous HMM. How can they achieve such an improvement? Is the speech data general and long enough to get this conclusion? How much additional extra computation will be needed to use this method? Is this method practical in a large speech recognition system? Here is a review of the paper “Multi-group Mixture Weight HMM” by L. Ming and Y. Tiecheng published in the Proceedings of the 6th International Conference on Spoken Language Processing, October, 2000 [1].

1. INTRODUCTION

Hidden Markov Model [2,3] (HMM) approach is one well-known and widely used statistical method of characterizing the spectral properties of the frames of a pattern [5,6]. In speech recognition, the underlying assumption of the HMM is that the speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be estimated in a precise, well-defined manner. As we know, speech signal can be characterized by a hidden state sequence. A Hidden Markov Model can be completely characterized by a matrix of state probabilities, observation densities and initial state probabilities. Study of these probabilities has found that the observation densities are most important for the performance of those recognizers using HMMs. Most improvement of HMM is made on this respect. In continuous observation densities Hidden Markov Model method, the observation densities for each state in the hidden state sequence is described by the mixture of weighted Gaussian density functions. Generally if we want to improve the performance of recognizer, a straight forward way is to use more Gaussian density functions for each state. Because logarithmic and exponential computation are very time-consuming, it will take much more time for training and recognition than the HMMs with fewer Gaussian density functions. Is there other way to describe a hidden state in Hidden Markov Model more precisely given a certain number of Gaussian density functions without increasing the time for training and recognition significantly? Multi-group Mixture Weight HMM is such a method that is announced by the authors to improve the performance with little additional computation.

By observing the re-estimate equations [2,4] carefully, 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 are modeled improperly. In this paper, each state has several groups of component weights. So it can meet different cases.

2. ALGORITHM REVIEW

The method can be briefly described as follows. Because of the importance of initial parameters for continuous density HMM, the first issue is to get proper initial component weights. At initial stage, for each frame in an speech signal utterance a vector called mixture weight vector is calculated and stored in the vector pool of the corresponding state. Then all mixture vectors in the vector pool of one state are classified into several clusters. The center vector of each cluster corresponds to a group of initial mixture weight. After initialization continuous density HMM can be trained by EM algorithm [7,8], which, of course, involves every group of mixture weight. For speech recognition, an appropriate group of mixture weight is selected by calculatig and comparing the output probabilities.

2.1. Retrieval of Initial Multi-group Mixture Weight

Because the initial parameters are essential to the quality of the final continuous density HMMs [2], the paper tries to get proper initial parameters of multi-group mixture weights for HMMs. It introduces Equ.(1), a widely used equation in continuous density HMM [4]. In (1), $\xi_t(j, k)$ is the probability of being in state j with the kth mixture components at time t, given the model λ and observation X.

$$\xi_t(j, k) = f(s_t = j, k_t = k | X, \lambda) \quad (1)$$

For the observation vector X_t at time t and given the model λ, the following variables are defined,

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

$$F_j(X_t) = \left(\frac{\xi_t(j, 1)}{\eta_t(j)}, \ldots, \frac{\xi_t(j, M)}{\eta_t(j)}\right) \quad (3)$$

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

After observing the initial weights, the authors find there are some defects in these weights since some weights are so high nearly to 1 while some weights are almost zeroes. This would degrade the real performance of the continuous density HMM. The authors decide to solve this problem by smoothing the mixture weights as shown below.

A group of mixture weight $F = \{c_1, c_2, \ldots, c_M\}$ can be smoothed by Equ.(4),

$$c_i' = c_i \cdot \theta + (1 - c_i) \cdot (1 - \theta) \quad (4)$$

where,

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

and $1 \le i \le M$.

After the above transformation, c_i' should be normalized by their sum.

Let,

$$C' = \sum_i c_i' \quad (6)$$

$$c_i'' = c_i'/C' \qquad 1 \le i \le M \quad (7)$$

Then the group of initial mixture weight will be

$$F = \{c_1'', c_2'', \ldots, c_M''\}.$$

2.2. Training of Multi-group Mixture Weight HMM

The observation vector X_t, which is assigned to the state j of the model λ, is given during the stage of training. Thus Each group of mixture weight and the corresponding output probability can be calculated. The mixture weight vector which gives the maximum output probability can be noted down. After each iteration, the pth group of mixture weight of state j, model λ for the current iteration can be got by adding up all mixture weight vectors which belong to the pth group, state j and model λ and dividing the sum by the number of vectors. The re-estimate formula for the mixture weight can be written as (8). From the output probability the best suitable group of mixture weight will be chosen.

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

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 [9]. After training and clustering these groups of mixture weight vector, recognition using these mixture weight vectors can be implemented.

2.3. Recognition using Multi-group Mixture Weight

After getting the groups of mixture weight, there is no much difference between the traditional HMM and multi-group mixture weight HMM for recognition. From calculations of the output probability of every observation with every group of mixture weight, the group which outputs the maximum probability would be chosen.

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

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

The authors announce here that when calculating the output probabilities, the probabilities of the Gaussian density functions can be calculated first, then they are multipled with each group of mixture weight respectively. They also claim that only several extra multiplication operations for each state are need and compared with the exponential operations, this additional computation can be ignored.

2.4. Theoretical analysis for Multi-group Mixture Weight HMM method

If we carefully examine the algorithms shown above, we can find there are some unresolved theoretical issues in this new method. First I notice that ε in Equ. (5) is a variable for which there is not any reference or explanation about what value it should be determined. Is it a random value? Why does it have a range from 0.5 to 0.7? What's the theoretical reference for this variable? Obviously the authors do not explain it clearly. Second issue is that given the transformed and normalized group of initial mixture weight, how can they get p groups of mixture weight, state j and model λ after Viterbi decoding? How can they classify those groups of mixture weight? What's the classification standard when they tries to cluster those groups of mixture weight? Third, in evaluating the additional time consumed by this new method, neither theoretical equations nor experimental data are provided for supporting the authors view. Several simple explanation sentences are not enough to assure reader about the time efficiency of this new method. More details about these important issues are needed in this paper and consequently we have reasons to throw doubts on theoretical conclusion of the authors.

3. EXPERIMENT AND ANALYSIS

The 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, the authors also made experiments with the traditional HMM which only has one group of mixture weight for each state. In their Multi-group Mixture Weight HMM, three groups of mixture weight for every state are used. modeling methods.

Table 1: Experiment results for the performance of the traditional HMM and Multi-group Mixture Weight HMM

	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%

From Table 1, 12% error reduction in total digital error rate is shown by the authors when three groups of mixture weight are used. There are several problems in these experiment data. The most obvious problem is that the description for test data is too simple. What is the test data like? Where can the readers find these digital strings? Are these digital strings spoken in English or Chinese? Insufficiency of detailed descriptions for test data makes it very difficult for readers to evaluate the real value of these experiments and this Multi-group Mixture Weight HMM approach. Supposing these digital strings were spoken by English, then the error rates for the two system are too high to be acceptable. On the other hand, 12% relative error rate improvement sounds good but after simple calculation we can find that only 0.32% error rate improvement is actually achieved for the whole error rate, which is actually not so great improvement.

4. SUMMARY

In traditional continuous density HMM, each state is characterized by the mixture of a few weighted Gaussian density functions. The weight coefficients for Gaussian density functions will be reestimated during the iterations of EM training [7,8] and then will be fixed for one state when they are used in the process of recognition. The main idea of this paper is to construct more than one group of mixture weight for one single state, which will give out more possible expressions for that state. After passing through training step, those fixed weight coefficients for one single state will be used respectively in recognition to give out different output probabilities. Then the group of mixture weight which gives out the maximum probability will be chosen as the best description for that state. But two important problems arise when this method is evaluated as being practically used in large speech recognition system. One is that how much on earth this method would improve the performance for state-of-art large speech recognition system. Since the weight coefficients for Gaussian density functions will also be reestimated during training and a certain number of Gaussian functions will be used for one state, the improvement involved by using this method may be ignored. The experiment data shown in the paper are too inefficient to show the readers that this is not a big problem. Another problem is that keeping more groups of mixture weight for one state means that the time used for recognition computation would be increased significantly. In large speech recognition system it would not be practical to use this method unless further theoretical demonstration and experiments show enough evidence of its practical value.

REFERENCES

[1] L. Ming, Y. Tiecheng, “Multi-group Mixture Weight HMM,” Proceedings of the 6th International Conference on Spoken Language Processing, vol. 1, pp. 290-292, Beijing, China, October 2000.

[2] L.R. Rabiner, B.H. Juang, “Fundamentals of Speech Recognition,” Published by PTR Prentice-Hall Inc., 1993.

[3] J. Picone, “Continuous Speech Recognition Using Hidden Markov Models,” IEEE ASSP Magazine, vol. 7, no. 3, pp. 26-41, July 1990.

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

[5] X.D. Huang, et al., “The SPHINX-II Speech Recognition System: An Overview,” Computer Speech and Language, vol. 2, pp. 137-148, February, 1993.

[6] 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, pp. 41-44, Detroit, Michigan, U.S.A, May 1995.

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

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

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