International Conference on Spoken Language Processing, pp. IV-141~144, Oct. 16-20, Beijing

AN EQUIVALENT-CLASS BASED MMI LEARNING METHOD FOR MGCPM

Chunhua Luo, Fang Zheng, Mingxing Xu
Center of Speech Technology, State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science & Technology, Tsinghua University, Beijing, 100084, China
LuoCH@sp.cs.tsinghua.edu.cn

ABSTRACT

In this paper, we present an Equivalent-Class Based Maximum Mutual Information (ECB-MMI) learning method for our previously proposed Mixed Gaussian Continuous Probability Model (MGCPM). Similar to HMMs, the defined object function for MGCPM training considers the mutual information among different models so as to maximally separate the Speech Recognition Units (SRUs) in model space. Experimental result shows that for MGCPM the MMI training method can improve the recognition rate by 5% compared to the traditional training method MLE (Maximum Likelihood Estimation). Because the computation amount of MMI algorithm is very large, we propose an N-Best strategy to find the corresponding equivalent class (EC) in order to reduce complexity. Our experimental result shows that this criterion works very well.

Keywords: Equivalent Class Based-MMI, Mixed Gaussian Continuous Probability Model, Speech Recognition Unit, MLE, and Equivalent Class

1. INTRODUCTION

It is well known that Hidden Markov Models (HMMs)[1] are popular in recent speech recognition. They include continuous mixture density HMMs [2] with full covariance matrices or diagonal covariance matrices, semi-continuous HMMs, and VQ-based discrete HMMs [3].

A continuous HMM (CHMM) is represented by the state transition probability matrix A, the observation probability density function matrix B and the initial probability distribution vector π. In our research, we found that the state transition probability matrix in a HMM is not significant. This motivates us to propose a new model named MGCPM [4] to overcome the shortcomings inherent in the conventional HMM, one is the inaccurate modeling of state duration while another is the inaccurate assumption of the conditional independence of observations given the state sequence. The MGCPM eliminates the state transition matrix while using the mixed Gaussian densities to described the intra-state feature spaces. The state transition is controlled by a non-linear segmentation (NLS) algorithm in the initial training step and a modified Viterbi algorithm or a frame synchronous network search algorithm both in the iterative training steps and the recognition procedure. Compared to the Continuous Densities HMM, MGCPM achieves a fast recognition speed with only a little loss in recognition rate.

Many algorithms are developed to estimate the HMM parameters, for example Maximum Likelihood Estimation (MLE) method, Maximum Mutual Information (MMI)[5][6]. The MMI method was firstly introduced by L.R. Bahl, but it is not very straightforward for MGCPM. When applied to MGCPM, the MMI training procedure should be modified.

This paper begins with a review of Maximum Likelihood Estimation. In the third section, we propose an Equivalent-Class based MMI learning method for MGCPM, including the object function and formula deduction etc. Then the experiment results are reported in Section 4. We also give our conclusion according to this experiment.

2. MLE

Normally a speech recognition system has two primary parts: the acoustic model and the language model. Suppose the input of the acoustic model, or the utterance is A, and the output of the acoustic model which is also the input of the language model, or possible word strings, is W, the system's task is to find the most likely work string W^* which satisfies the following equation

$$W^* = \arg\max_W P(A|W)P(W) \tag{1}$$

This equation is derived from the Bayesian rule. Because $P(A)$ is a prior probability that is not a function of the string W, it dropped from the maximization. $P(A|W)$ is the conditional probability of the utterance on the specific word string given by the acoustic model while $P(W)$ is the word string probability given by the language model. While the language model is obtained independently on the acoustic model. So the acoustic modeling process is aimed to specify the model that better describe the given training data. In order to get better performance in speech recognition process when using formulation (1), many techniques are presented in parameter estimation. MLE is just one of them.

MLE (Maximum Likelihood Estimation) attempts to maximize the likelihood of generating the training data with the right model. Its object function can be expressed as follows:

$$h(\theta) = p(Z|A(\theta)) \tag{2}$$

where Z is the output feature vector set of a acoustic model, $Z = \{z_j : j = 1,2,...,J\}$. A denotes the corresponding acoustic model of these feature vector and θ is the model parameter.

In our MGCPM, we use the following probability density functions (pdfs) to describe the distribution of feature vector z_j

$$p(z_j \mid \theta) = \sum_{m=1}^{M} \{ g_m p(z_j \mid \theta_m) \} \quad (3)$$

Mean vector u_m and covariance matrix R_m make up of the parameter θ_m of the m th Gaussian distribution. g_m is the weight of the m th Gaussian distribution.

In parameter estimation process, we make an assumption that z_j obeys the mixed Gaussian distribution and the vectors in Z are independent on each other. Based on these, the object function (2) can be rewrote as

$$h(\theta) = p(Z \mid \theta) = \prod_{j=1}^{J} p(z_j \mid \theta) \quad (4)$$

Taking the derivative of $h(\theta)$ with respect to parameter θ. When $h(\theta)$ reaches its extremum, the following equation will be satisfied

$$\nabla_\theta (\ln(p(Z \mid \theta) = 0 \quad (5)$$

For simplicity, we define

$$P_{mj} = p(z_j \mid \theta)^{-1} g_m p(z_j \mid \theta_m) \quad (6)$$

From Equation (5) we obtain the iteration formula of u_m, R_m, g_m as follows

$$\hat{u}_{md} = \sum_{j=1}^{J} (P_{mj} . z_{jd}) \Big/ \sum_{j=1}^{J} P_{mj} \quad (7)$$

$$\hat{R}_{md} = \sum_{j=1}^{J} (P_{mj} . z_{jd}^{\ 2}) \Big/ \sum_{j=1}^{J} P_{mj} - \hat{u}_{md}^{\ 2} \quad (8)$$

$$\hat{g}_m = J^{-1} \cdot \sum_{j=1}^{J} P_{mj} \quad (9)$$

It is well known that if the true distribution of the data lies in the space of the assumed distribution and training data are sufficient, the pdfs parameterized by the ML estimates will converge to the true distribution of the data.

Unfortunately, these conditions can hardly be satisfied in real speech recognition systems, so L.R. Bahl proposed the MMI training method for HMM, but it can't be used for MGCPM training directly. In the following section, we will propose an equivalent-class based MMI learning method for MGCPM which is proved to be high efficient by experiment results.

3. MMI FOR MGCPM

During the training process using MLE, we only consider the object model and do not pay attention to the interference introduced by other models. As a result, the model parameter of our speech recognition unit (SRU) will overlap greatly in model space and decrease the model description ability. An alternative method to MLE is the MMI method. It attempts to maximize the discrimination between the correct model and the incorrect models.

3.1. Object function

During the MMI training process, it not only considers the model what the training data belongs to, but also takes into account the interactions among different models. The training process is to maximize the probability of the focused model given the acoustic observation sequence, and minimize the probability of other models. Models trained by this method can better describe the feature space. The object function of MMI can be stated as follows:

$$h(\theta) = p(A(\theta) \mid Z) \quad (10)$$

Using probability theory, we obtain the following equation

$$h(\theta) = \frac{p(Z \mid A(\theta)) p(A(\theta))}{\sum_{A'} p(Z \mid A'(\theta')) p(A'(\theta'))} \quad (11)$$

where A' is the acoustic model corresponding to a SRU and θ is the model parameter which we have already mentioned in Section 2. The value of $p(A(\theta))$ is a prior probability. In our experiment, we make a simplification that this value can be derived from the statistics of training corpus.

3.2. Parameter estimation

If we assume the feature vectors in Z are independent, then the object function (10) can be rewritten as follows

$$h(\theta) = p(A(\theta) \mid Z) = \prod_{j=1}^{J} p(A(\theta) \mid z_j) \quad (12)$$

when $p(z \mid \theta)$ reaches the maximum value, $\nabla_\theta (\ln(p(Z \mid \theta) = 0$ must be satisfied. Accordingly we obtain the following formula deduction procedure.

1.formula deduction of u_m

Taking the deriative of $\ln(h(\theta))$ with respect to parameter θ, we will have

$$\nabla_{u_m} (\ln(p(Z \mid \theta)) = \sum_{j=1}^{J} ((p(z_j \mid \theta) p(\theta))^{-1} - (\sum_{n=1}^{N} p(z_j \mid A_n) p(\theta_n))^{-1}) g_m p(z_j \mid \theta_m) R_m^{-1} (z_j - u_m) \quad (13)$$

For simplicity, we will introduce two factors P_{mj} and P_j.

$$P_{mj} = ((p(z_j \mid \theta) p(\theta))^{-1} - (\sum_{n=1}^{N} p(z_j \mid A_n) p(\theta_n))^{-1}) g_m p(z_j \mid \theta_m) \quad (14)$$

$$P_j = (p(z_j \mid \theta_m) p(\theta_m))^{-1} - (\sum_{n=1}^{N} p(z_j \mid A_n) p(A_n))^{-1} \quad (15)$$

$\hat{u}_{md}$ can be represented by the following iterative equation.

$$\hat{u}_{md} = \sum_{j=1}^{J} (P_{mj} . z_{jd}) \Big/ \sum_{j=1}^{J} P_{mj} \quad (16)$$

Here $\{A_n : n = 1, 2, ... N\}$ is the set of all possible acoustic model. In our speech recognition system, we choose Chinese syllables as our speech recognition units, and the number of units is $N = 397$.

2. Formula deduction of

$$\nabla_{R_m}(\ln(\ p(Z\mid\theta)))$$
$$=\sum_{j=1}^{J}P_j\nabla_{R_m}(\sum_{m=1}^{M}g_m\,p(z_j\mid\theta_m)) \qquad (17)$$
$$=0$$

According to above equation, we can obtain that

$$\hat{R}_{md}=\sum_{j=1}^{J}(P_{mj}\cdot z_{jd}^{\ 2})\Big/\sum_{j=1}^{J}P_{mj}-\hat{u}_{md}^{\ 2} \qquad (18)$$

3.Formula deduction of g_m

Because g_m must satisfy that $\sum_{m=1}^{M}g_m=1$, so this is a conditional extremum. By applying Lagrange operator, we will have the following expression:

$$\sum_{j=1}^{J}\ln(P_j\sum_{m=1}^{M}g_m\,p(z_j\mid\theta_m))+\lambda(\sum_{m=1}^{M}g_m-1) \qquad (19)$$

Take the deriative of above expression with respect to parameter g_m,λ, we will have

$$\begin{cases}\sum_{m=1}^{M}g_m=1\\ \sum_{j=1}^{J}P_j\,p(z_j\mid\theta_m)+\lambda=0, m=1,2,...,M\end{cases} \qquad (20)$$

we get that

$$g_m=-\sum_{j=1}^{J}P_{mj}\Big/\lambda\ ,\ \lambda=-\sum_{m=1}^{M}\sum_{j=1}^{J}P_{mj} \qquad (21)$$

hence , finally

$$g_m=\sum_{j=1}^{J}P_{mj}\Big/\sum_{m=1}^{M}\sum_{j=1}^{J}P_{mj}, m=1,2,...,M \qquad (22)$$

3.3. Implement of the algorithm

Using Equations (16), (18), and (22), we can get the iterative training process for MMI algorithm. It is something like the gradient descent method. Following are the outline of MMI training process.

Step1: Training the initial models using MLE algorithm;

Step2: Calculate the prior probability $p(\theta)$ according to training corpus;

Step3: Training models using Equations (16), (18), and (22);

Step4: If not convergent, go to Step3 and continue the iterative process else exit this procedure.

Many factors can influence the final mode parameters obtained via the MMI algorithm, for example, the quality of our training sample, the algorithm complexity, the iteration control strategy and the update method for model parameter. Among all these factors, the algorithm complexity is the most important thing we should pay attention to. In the MMI training process, it involves the procedure that requires the computation of the likelihood of the acoustic observation sequence given all possible models; this is easily done for small-vocabulary systems, but not as easily done for large-vocabulary systems. Though it is a very useful training algorithm, it can hardly be applied into real speech recognition system without the decrease of computation amount.

Based on this consideration, we applied the equivalent-class into the training process to reduce computation amount.

3.4. Using equivalent-class

In fact, the number of the speech recognition units that may interfere the focused training unit is not the same large as, or even much smaller than, the total number of units. Hence in the definition of the MMI training object function there is no need to consider all the units, but only those similar units. For each unit to be trained, there is one corresponding equivalent class (EC), i.e., a confusion set, which is the only part we consider in the ECB-MMI method. An N-Best strategy is used to find the EC for each unit, according to the model distance matrix or the recognition matrix. Experimental result shows that these two criteria both work well.

For the Gaussian distribution, if we define the overlap area of two single variance Gaussian as follows

$$O(g^{(1)}(\cdot),g^{(2)}(\cdot))=\int_{x=-\infty}^{+\infty}\min(g^{(1)}(x),g^{(2)}(x))dx \qquad (23)$$

then the distance between the Gaussian distribution with diagonal covariance matrix can be formulated as

$$D(g^{(1)}(\cdot),g^{(2)}(\cdot))=\sum_{i}-\log O(g_i^{(1)}(\cdot),g_i^{(2)}(\cdot)) \qquad (24)$$

By calculating the distance between each SRU, we can obtain the model distance matrix.

Unfortunately, even though this method is very accurate, the model-similarity computation amount is also something undesirable for us. So in our experiment, we use another method to find the EC for each unit.

The number of SRUs is often greatly larger than the number of SRUs in a confusion set. By analyzing the recognition result, we find an import phenomenon that some confusion SRUs take great part of error samples in total samples of all the confusion SRUs. We can use these SRUs to make up of our equivalent class. The following table can illustrate this point.

Table1 is obtained from our experiment where we choose syllable /*shang*/ as an example. The sample number of /*shang*/ we used in experiment is 434; we find the number of the error recognized samples is 130. In these samples, the most frequently occurred error sample is /*zhang*/ which almost takes up to 20% error samples in total samples.

Table1 Error sample distribution of /*shang*/

Unit	Number	Unit	Number	Unit	Number
zhang	27	*sha*	26	*Chang*	20
Sheng	12	*sa*	5	*shao*	5
shi	4	*dang*	4	*xiao*	3
jia	3	*zheng*	2	*ba*	2
Dao	1	*san*	1	*she*	1

We also find that the number of samples that greatly influence the recognition result is much small. In the above table, only about 8 samples that we should pay attention to. Based on this phenomenon, we can use another N-Best strategy to find the equivalent class of each unit, that is the use of recognition matrix. We only need to count the number of error samples by analyzing the recognition result and sort the number to choose the most influential SRUs. The value of *N* can vary according to a specific purpose (for example, computation complexity or model description ability. Obviously, the more large the N is, the more large the computation complexity will be, and the better the model description ability would be). We choose 10 in our experiment.

All the N-best candidates will make up the recognition matrix in which one element stands for one unit in EC. It's obvious that the computation amount will reduce greatly by applying this matrix to the MMI training process. For example, if we choose *N*=15,then the computation amount in likelihood computation will decrease by (397-15)/397=96%. It brings out approximate 90% overall computation amount reduction.

4. EXPERIMENTAL RESULTS

The experiment is made across a continuous Mandarin speech database recorded by 38 males. Each speaker uttered one set of sentences in a continuous mode. The database contains 250,657 Mandarin syllables totally. We used 30 males' utterances to train the MGCPMs. The remaining part is used for testing. All the recorded materials were obtained in an officelike environment through a close-talk noise-canceling microphone. They are digitized at a sampling frequency of 16KHZ. A 32ms Hamming window is performed on each frame of the speech. And then the cepstral coefficients derived from LPC of order 16 are extracted for every 16ms.The acoustic model used for experiment is MGCPM with 6 states and 16 mixtures each state.

Table 2 gives the experiment results using MMI, ECB-MMI, and compares it with the results using MLE.

Table 2. Comparison of three algorithms

Method	**1**	**2**	**3**	**4**	**5**
MLE	72.01	78.30	81.66	83.91	86.57
MMI	76.77	83.15	86.68	89.01	91.74
ECB-MMI	75.32	81.97	85.55	87.89	90.68

Table 2 shows the hit rates of top 5 candidates. The results indicate that the performance has been raised by about 5% when using the MMI method and raised by 3~4 percent when using ECB-MMI, compared with the traditional MLE. The ECB-MMI has a great deal of reduction in computation complexity, which is the most preferable aspect of this learning method.

The result shows that the enhancement of hit rates when using MMI indeed owe to the consideration of other SRUs that may interfere the training process of the focused SRU. And the more economical algorithm ECB-MMI can lead to 90% reduction in computation amount but only with about 2% recognition rate reduction compared to MMI for MGCPM.

We have already pointed out that MLE is not so desirable is partly because of the insufficient training data. This is proved by our experiment. We design an experiment where the model mixture is 8 for each state. It can be regarded as the increase of training data. We found that MMI algorithm does not have too much advantage compared with MLE algorithm because the increase of training data. In this situation, the hit rate of MMI algorithm only increased about 1%. But if we do not have sufficient training data, the MMI can overcome this shortcoming by using the mutual information. That is to say, the MMI algorithm is a good learning method for us to choose.

5. CONCLUSIONS

This paper has reported an effective equivalent-class based learning method for MGCPM and introduced the N-best strategy that finds the confusion set, i.e., equivalent-class (EC). The N-best strategy based on a recognition matrix is shown to be very effective.

Experimental results show that ECB-MMI gives considerable reductions in recognition error rate for speech recognition unit and greatly reduces the computation amount.

In the experiment, we find that the recognition rates of some SRUs are not improved as much as we expected. This needs us to make an overall analysis on the training procedure. Maybe it is partly due to some bad samples of the training data; maybe the iterative procedure of MMI should be modified; or maybe the N-Best strategy to find the EC is not so good. All these need us to do further research in the future.

6.REFERENCES

1. L.R.Rabiner(1989),“ A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition”, *Proc. IEEE*, 72(2):257-286, Feburary 1989

2. B.H.Juang,L.R.Rabiner,(1985) “Mixture autoregessive hidden Markov Models for speech Signals,” *IEEE Trans. On Acoust.,Speech,and Signal Processng*, 1985, 33:1404-1413

3. L.R.Rabiner,S.E.Levinson,M.MSondhi,(1983) “On the application ofvector quantization and hidden Markov models to speaker-independent,isolated word recognition,”*Bell Syst.TechJ.*,1983,62:1075-1105

4. X.L.Mou,Q.X.Hu,W.H.Wu,(1997) “An Context-Independent strategy Speaker Verification System.”, *Journal of Tsinghua Univ.*1997,3,vol 37

5. Y. Normandin,R. Lacouture, R. Cardin，“MMIE Training for Large Vocabulary Continuous Speech Recognition”，Proc. ICSLP'94, Volume 3, pages 1367-1370, Yokohama, September 1994

6. L.R.Bahl, P.Brown, P.de Souza, R.Mercer(1986), “Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition”, *ICASSP86*, Tokyo, vol1, pp.49-52