Speaker Recognition Based on SOINN and Incremental Learning Gaussian Mixture Model

Zelin Tang, Furao Shen and Jinxi Zhao

Abstract— **Gaussian Mixture Models has been widely used in speaker recognition during the last decades. To deal with the dynamic growth of datasets, initial clustering problem and achieving the results of clustering effectively on incremental data, an incremental adaptation method called incremental learning Gaussian mixture model (IGMM) is proposed in this paper. It was applied to speaker recognition system based on Self Organization Incremental Learning Neural Network (SOINN) and improved EM algorithm. SOINN is a Neural Network which can reach a suitable mixture number and appropriate initial cluster for each model. First, the initial training is conducted by SOINN and EM algorithm only need a limited amount of data. Then, the model would adapt to the data available in each session to enrich itself incrementally and recursively. Experiments were taken on the 1st speech separation challenge database. The results show that IGMM outperforms GMM and classical Bayesian adaptation in most of the cases.**

I. Introduction

MAXIMUM LIKELIHOOD (ML) estimation and maximum posteriori estimation were applied in most of speaker recognition systems based on Gaussian mixture models (GMM) [1]. It had been previously demonstrated that robustness, performance and the best trade-off in terms of complexity were shown in systems based on Mel-frequency cepstrum coefficients (MFCC) and Gaussian mixture models. In this paper, we follow this framework and focus on further enhancing the performance, adaptability and incremental learning of the GMM model.

D. A. Reynolds [2] et al. proposed Gaussian mixture model on the basis of probability statistics against the field of speaker recognition.

Speaker recognition system based on GMM are extracting feature vectors from short time speech frames and simulating topology structure by a set of multidimensional Gaussian probability density. It solves the problem of short-time speech and text independent speaker recognition.

For speaker recognition, there had been numerous algorithms presented in literature [3]-[5]. There are many commonly used techniques such as maximum a posteriori (MAP) adaptation [6], large margin gaussian mixture models (LMGMM) [7], split and merge EM algorithm [8] and maximum likelihood linear regression (MLLR) adaptation [9]. Gaussian mixture model is one of the famous discriminative approaches, which recently attracts significant attention.

To deal with the dynamic growth of datasets, initial cluster problem and achieving the clustering results effectively on incremental data, an incremental adaptation method called incremental learning Gaussian mixture model (IGMM) is proposed in this paper. It was applied to speaker recognition system based on self organization incremental learning neural network (SOINN) [10] and improved EM algorithm.

Our GMM learning procedure starts from an initial estimate of the parameters and uses Expectation-Maximization (EM) algorithm to estimation and maximum a posteriori estimation. However, the EM algorithm has been known to suffer from several problems. One of the problems is that, as a local method, it is too sensitive to the selected initial parameter estimates, and it may converge to the boundary of parameter leading to inaccurate estimation [11]. Another problem is that the mixture number of a specific speaker GMM is not known in advance and one speaker's mixture number is likely to be different from another. Therefore we present a way to improve these defects, which make the model insensitive to initial parameter and improve itself continuously in incremental learning. Our way is:

First, using SOINN instead of k-means to generate primary cluster in initial learning. Second, we give the GMM some prior limited size clusters and let the system adjust the mixture number in incremental learning. With the prior limited size, a new cluster will be composed when some samples don't belong to any cluster in GMM, called external sample. Third, we collect these external samples to form some new clusters to perfect the spatial distribution of GMM. This improved algorithm is called adaptive EM, which is used in incremental learning of GMM.

Section 2 will give the scheme of GMM incremental learning, and in section 3, the proposed algorithm were evaluated. At last, in section 4, we will draw some conclusions.

II. Incremental Learning of GMM

A. Modified MFCC and Frame Likelihood Score

MFCC can be regarded as standard feature for speech and speaker recognition systems. In our experiment, the speech signals sampled at $16kHz$ are divided into equal length frames by overlap windows. The each frame, with length of $16ms$ and overlap of $8ms$, is converted into MFCC vector. MFCC are calculated by the triangular filter bank procedure [12], with the pre-emphasis factor of 0.95 in experiment. We remove mute segment in speech with a way of voice activity detection (VAD) based on teager energy [13] and power

Zelin Tang, Furao Shen and Jinxi Zhao are with the National Key Laboratory for Novel Software Technology, and Department of Computer Science and Technology, Nanjing University, China.(email:tangzelin@163.com,{frshen,jxzhao}@nju.edu.cn)

This work was supported in part by the 973 Program 2010CB327903, the Fund of the National Natural Science Foundation of Jiangsu NSF grant BK2011567.

spectrum variance [14]. Because the active segment has higher quality than mute segment. And it was confirmed by experiment that higher recognition rate could be represented using active segment instead of the whole speech. MFCC is extracted as the figure below

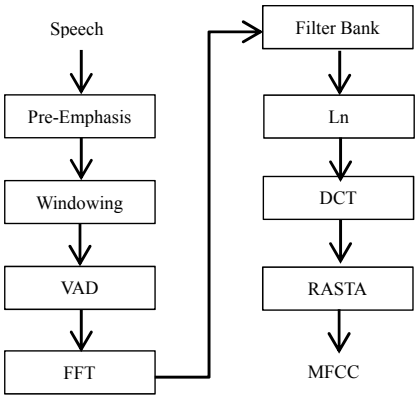

Fig. 1. The process of extracting MFCC

First, conduct Fourier transform for each frame, as in (1).

$$X_i(\omega) = FFT(S_i(n)) \tag{1}$$

where $i \in 1, 2, \ldots, I$ refers to frame index, $S_i(n)$ refers to sampled data of frame i, $n \in \{1, 2, \ldots, N\}$ donates nth sampled point of a frame.

In windowing process, 256 sample points each frame. Then, we weight the square of power spectrum with square of frequency, as in (2):

$$f_{i,k} = \omega_k^2 |X_i(\omega_k)|^2 \tag{2}$$

where ω_k donates frequency. The teager energy of frame i as follows:

$$T(i) = \sum_{k=1}^{n} (f_{i,k})^{1/2} \tag{3}$$

Second, the power spectrum variance is calculated on the basis of power spectrum mean μ_i, as in (4):

$$\mu_i = \frac{1}{K} \sum_{k=1}^{K} |X_i(\omega_k)| \tag{4}$$

The power spectrum variance as follows:

$$S(i) = \sum_{k=1}^{K} (|X_i(\omega_k)|^2 - \mu_i)^2 \tag{5}$$

where $|X_i(\omega_k)|$ donates power spectrum of frame i. We use teager energy and power spectrum variance to filter out some poor quality speech frames, so that high quality model and high recognition rate will be shown. In experiment, a threshold of power spectrum variance $S(i)$ is $S_{Threshold} = 0.01$.

Third, teager energy is used to revise segment boundary of active frame. If a frame i close to active frame segment and satisfy the inequality $T(i) > T_{Average} * 0.6$, the frame i will be added to the active frame segment. $T_{average}$ donates the mean value of an active frame segment.

Fourth, the Mel cepstrum filtering as follow:

$$m(l) = \sum_{k=o(l)}^{h(l)} W_l(k) |X(k)| \tag{6}$$

$$W_l(k) = \begin{cases} [k - o(l)]/[c(l) - o(l)] & o(l) < k \leq c(l) \\ [h(l) - k]/[h(l) - c(l)] & c(l) < k \leq h(l) \end{cases} \tag{7}$$

where $c(l), o(l)$ and $h(l)$ is respectively low, middle and high of the lth triangle filter, $m(l)$ donates filtering energy of lth filter. The discrete cosine transform as follows:

$$MFCC_i = \sqrt{(\frac{2}{N}) \sum_{l=1}^{L} \{log[m(l)] cos[(l - \frac{1}{2}) \frac{i\pi}{L}]\}} \tag{8}$$

where $l \in \{1, 2, \ldots, L\}$ is the number of triangle filters, $L = 48$ in experiment.

Fifth, in order to achieve better results, we apply the combination of 20-dimensional MFCC coefficient and 20-dimensional first-order differential MFCC coefficient as the feature parameter, as in (9):

$$\triangle MFCC_i = \frac{MFCC_{i+1} - MFCC_{i-1}}{2} \tag{9}$$

Relative Spectra (RASTA) technology is a method of inhibiting channel noise. It can effectively inhibit channel distortion, making system robust to various channels. The RASTA filter as follow:

$$H(Z) = G \times \frac{Z^{N-1} \sum_{n=0}^{N-1} [(N-1)/2 - n] Z^{-n}}{1 - \rho Z^{-1}} \tag{10}$$

with $N = 5, G = 1$ and $\rho = 0.94$.

B. Traditional Gaussian Mixture Model

In speaker recognition, there are two types of modeling methods: the deterministic methods, dynamic time warping (DTW) and vector quantization (VQ); and statistics methods, Gaussian mixture model (GMM) and hidden Markov model (HMM). Describing speech feature density distribution based on feature space by probability density function should meet some basic conditions, for example, describing the distribution of voice feature, simple and common function, easy to estimate parameter and train. Gaussian mixture model achieve these basic conditions. Gaussian mixture model consists of M component densities, as in (11):

$$p(x) = \sum_{m=1}^{M} [\alpha_m g_m(x)] \tag{11}$$

where x is a D-dimensional random variable, $g_j(x)$ is the Gaussian probability density and α_j, $(j = 1, 2, \ldots, M)$, is

the mixture weights. Each component density include a D-dimensional Gaussian probability function with the form as in (12):

$$\begin{aligned} g_j(x) &= N(x, \mu_j, \Sigma_j) \\ &= \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right] \end{aligned} \tag{12}$$

with mean vector μ_j and covariance matrix Σ_j, and the mixture weights satisfy the constraint that $\sum_{m=1}^{M} \alpha_m = 1$. These three parameters: mean vectors, covariance matrices and mixture weights, constituting the complete Gaussian mixture density. These parameters are collectively represented by $\theta = \{\alpha_j, \mu_j, \Sigma_j | j = 1, 2, \ldots, M\}$. For speaker identification, each speaker is represented by a GMM and is referred to by his/her model θ.

In our work, as is in traditional GMM system, diagonal covariance was used for the purpose of preventing singular matrix in iterative process. It had been demonstrated that this method could make the model simplification and almost did not lead to loss of precision. We found this assume was valuable to simplify computation complicity with obtaining acceptable results. Equation (2) can be rewritten as:

$$g_j(x) = \prod_{d=1}^{D} \{\frac{1}{\sqrt{2\pi}\sigma_{jd}}\} \exp\left[-\frac{(x_d - \mu_{jd})^2}{(2\sigma_{jd}^2)}\right] \tag{13}$$

where σ_{jd}^2 is the dth diagonal element of diagonal covariance matrix σ_j^2, it is the same in the following.

C. Training of Traditional Gaussian Mixture Model

Maximum Likelihood (ML) based on Expectation Maximization (EM) algorithm had been applied to re-estimate the parameters of the GMM iteratively [12]. For the purpose of guaranteeing likelihood value increase monotonously, following re-estimation formulas were applied in EM iteration. Mixture Weights, as in (14), mean, as in (15), variance, as in (16), the posteriori probability for acoustic class j, as in (17):

$$\acute{\alpha}_j = \frac{1}{t}\sum_{i=1}^{t} \beta_j(x_i) \tag{14}$$

$$\acute{\mu}_j = \frac{\sum_{i=1}^{t} \beta_j(x_i) x_i}{\sum_{i=1}^{t} \beta_j(x_i)} \tag{15}$$

$$\acute{\sigma}_{jd}^2 = \frac{\sum_{i=1}^{t} \beta_j(x_i)(x_{id} - \mu_{jd})^2}{\sum_{i=1}^{t} \beta_j(x_i)} \tag{16}$$

$$\beta_j(x) = \frac{\alpha_j g(x; \mu_j, \sigma_j^2)}{\sum_{m=1}^{M} \alpha_m g(x; \mu_m, \sigma_m^2)} \tag{17}$$

D. Incremental Gaussian Mixture Model

Before training traditional GMM, the number of mixture components must be determined. It equals the value of K in K-means method. However, it is difficult to choose a suitable mixture number for every speaker. After the number was determined, it can not be modified unless the speaker model is re-trained entirely. This disadvantage makes us spend too much time to decide how many mixture components should be chosen since each speaker have different features and features of one speaker can change year by year. So we need SOINN and incremental learning. The advantage of SOINN is self-adaption, because it can give a suitable number of mixture components within a short time. As we all know, K-means method has a parameter K that needs to be determined, and it may appears non-convergence in 100-iteration in MATLAB. SOINN could solve these problems.

SOINN is a two-layer neural network. Learning the density distribution of the training data and representing it by nodes and edges are the purpose of the first-layer. The second layer can separate clusters by deleting the low-density clusters, and represent the topological structure of training data simplify. The detail algorithms of SOINN see literatures [10] and [17]-[19].

In this paper, we proposed an improved EM algorithm to train incremental GMM. Incremental learning means repeatedly training without destroying the old prototype patterns. Therefore, the EM iterative formula is modified to meet this requirement. The new method of re-estimation in each iteration step are fusion of both old and new data and generation of new cluster automatically, instead of weighted combination of old and new clusters.

By the experimental statistical analysis, glottal information characteristics of human basically fit normal distribution [15]. Weighted merging clusters will cause the model and data mismatch. Our method use mixture weight (α), mean (μ), variance (σ) and the feature vectors number (t) in each cluster to describe GMM. This will save storage and computational complexity.

The parameter t refers to the number all feature vectors in an incremental learning on the basis of the original model $\hat{\theta} = \{\hat{\alpha}, \hat{\mu}, \hat{\sigma}^2, \hat{t}\}$. The new iteration formula for mixture weight as in (18):

$$\acute{\alpha}_j = \frac{\sum_{i=1}^{t} \beta_j(x_i)/t + \hat{\alpha}_j}{2} \tag{18}$$

mean as in (19):

$$\acute{\mu}_j = \frac{t_j \sum_{i=1}^{t}[\beta_j(x_i)x_i] / \sum_{i=1}^{t} \beta_j(x_i) + \hat{t}_j \hat{\mu}_j}{\hat{t}_j + t_j} \tag{19}$$

variance as in (20)

$$\acute{\sigma}_{jd}^2 = \frac{\{t_j \frac{\sum_{i=1}^{t} \beta_j(x_i)(x_{id} - \mu_{jd})^2}{\sum_{i=1}^{t} \beta_j(x_i)} + \hat{t}_j \hat{\sigma}_{jd}^2 + \Theta_{jd}\}}{(t_j + \hat{t}_j)} \tag{20}$$

$$\Theta_{jd} = \frac{t_j \hat{t}_j (\mu_{jd} - \hat{\mu}_{jd})^2}{(t_j + \hat{t}_j)} \tag{21}$$

number of feature vectors, as in (22)

$$\acute{t_j} = t_j + \hat{t}_j \tag{22}$$

where x_i is the extracted feature vector at frame i, t_j donates the feature vectors number of cluster j. $\hat{\alpha}_j$, $\hat{\mu}_j$, $\hat{\sigma}_j$ and $\hat{t}_j$ are respectively the mixture weight, mean, variance and feature vectors number of cluster j in the trained GMM. α_j μ_j, σ_j and t_j are respectively the mixture weight, mean, variance and feature vectors number of cluster j in incremental learning. $\hat{\sigma}_{jd}$, $\hat{\mu}_{jd}$ (d=1,2,...,D) are the d dimension of $\hat{\sigma}_j$, $\hat{\mu}_j$. Here $\sum_{i=1}^{t}[\beta_j(x_i)(x_{id}-\mu_{jd})^2]/[\sum_{i=1}^{t}\beta_j(x_i)]$ equivalent to σ_{jd}^2, $\beta_j(x_i)$ is in the same way as (17).

In order to learn new cluster in incremental learning, we restrict cluster relative size. Equation (13) could be written as:

$$g_j(x_i) = \prod_{d=1}^{D}\{[\frac{1}{\sqrt{2\pi}\sigma_{jd}}]\exp[-\frac{1}{2}\Phi_j(x_{id})]\} \tag{23}$$

where x_{id}, σ_{jd} is respectively the d-dimension of x_i, σ_j, $\Phi_j(x_d)$ is defined as in (24):

$$\Phi_j(x_d) = \begin{cases} \frac{(x_d-\mu_{jd})^2}{\sigma_{jd}^2} & \text{if}(\frac{(x_d-\mu_{jd})}{\sigma_{jd}} < \lambda) \\ \infty & \text{elsewise} \end{cases} \tag{24}$$

If $\forall\ d \in (1,2,\ldots,D)$, $\Phi_j(x_d) < \lambda^2$, we take x is belong to the cluster j, otherwise not belong to. Moreover, if there is a feature vector x_i satisfy the conditions $\forall\ d \in (1,2,\ldots,D)$, $\Phi_1(x_{id}) < \lambda^2$ and $\Phi_2(x_{id}) < \lambda^2$, we consider x_i is the common feature vector of clusters m_1 and m_2. Two clusters which have a certain amount of common feature vectors will be merged into one. There had been an effective way to merge Gaussian clusters[19]. But that method splits and merges clusters through large number of calculations, and it is a non-incremental method. Therefore, we use the new iteration formula. First, this approach makes the model match with the original characteristic better after incremental learning. Then, this method makes cluster more consistent after merged. At last, the specific vectors were discarded when incremental learning finished, only representative parameters leaved.

In incremental learning, it is speaker model $\theta = \{\alpha_j, \mu_j, \Sigma_j, t_j | j = 1,2,\ldots,M\}$ instead of trained data will be saved after the end of learning. So we use speaker model parameter θ take the place of trained data to merge with incremental data in every step of iteration. This method could avoid parameters of incremental data mismatch with true cluster center in iterations. For example, we merge two Gaussian clusters with our method. Clusters $\{m_1|\mu_1 = (0,0), \sigma_1^2 = [10,0;0,10]\}$ and $\{m_2|\mu_2 = (9,0), \sigma_2^2 = [10,0;0,10]\}$ are two Gaussian distributions as the follow figure.

Cluster m_3 merged by the two clusters m_1 and m_2.

However, it is not true that any two clusters can be merged into one cluster. Density of cluster is calculated before and after merged. If cluster variances σ change too much after being merged, the merger would be canceled.

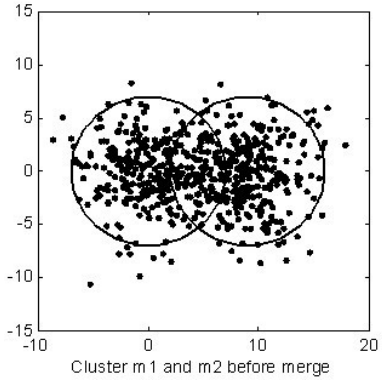

Fig. 2. From left to right respectively are the Gaussian distributions of m_1 and m_2 before merged

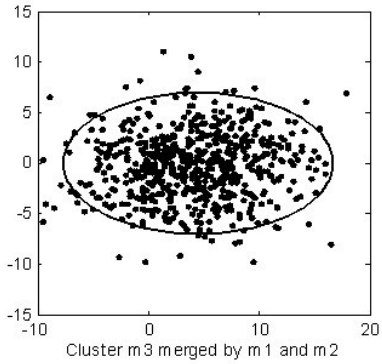

Fig. 3. The Gaussian distribution m_3 merged by m_1 and m_2

That is to say, when we find two clusters $\{m_1|\mu_1, \sigma_1^2, t_1\}$ and $\{m_2|\mu_2, \sigma_2^2, t_2\}$ have common feature vectors more than a certain percentage number, we try to merge this two clusters into cluster $\{m_3|\mu_3, \sigma_3^2, t_3\}$. If $\sigma_3 > \varphi * (\sigma_1 + \sigma_2)$, we will not merge m_1 and m_2. φ equals 1 in our experiment. The merge is defined as:

$$\mu_3 = \frac{t_1\mu_1 + t_2\mu_2}{t_1 + t_2} \tag{25}$$

$$\sigma_3^2 = \frac{t_1\sigma_1^2 + t_2\sigma_2^2 + [t_1t_2(\mu_1-\mu_2)^2]/(t_1+t_2)}{t_1 + t_2} \tag{26}$$

$$t_3 = t_1 + t_2 - t_{12} \tag{27}$$

where t_{12} is the common feature vectors number of clusters m_1 and m_2. However, if m_1 is trained cluster and m_2 is incremental cluster, there are no common feature vector in m_1 and m_2, because trained data are always different from incremental data.

In incremental learning, there are always some feature vectors out of all clusters limit. We call these vectors inferior-vectors. These inferior-vectors may be isolated noise or new features of speaker. On account of the lack of speech sample and the speaker voice changing year by year, the speaker model may not learn all characteristic of speaker. We add new clusters in incremental learning to solve this problem.

Considering the characteristic of noises scattered and speech features concentrate, we apply SOINN to cluster the inferior-vectors and estimate which are noises.

Cluster density ρ will be calculated after inferior-vectors are clustered by SOINN. We will select the clusters which have more feature vectors or cluster density than others. In experiment, the cluster $\overline{m}_j$ clustered by SOINN which have feature vectors number $\overline{t}_j$ and cluster density $\overline{\rho}_j$ satisfy the conditions, as in (28), would be added to the speaker model.

$$\begin{cases} \overline{t}_j > \frac{\sum_{m=1}^{\hat{M}} \hat{t}_m}{h * \hat{M}} \\ \overline{\rho}_j > \frac{\sum_{m=1}^{\hat{M}} \hat{\rho}_m}{\hat{M}} \end{cases} \tag{28}$$

where $\hat{M}$ is the number of clusters in trained speaker model, $\hat{\rho}_j$ donates the cluster density in trained model, h is a variable to adjust the learning rate and can be adjusted as learning times increased, h equal to 10 in the experiment. The cluster density is defined as follows:

$$\rho_j = \frac{\|\sigma_j^2\|_2}{t_j} \tag{29}$$

E. Test Method

In test experiment, the speeches are translated into MFCC, 40-dimensional vectors set $\{X|X = (x_1, x_2, \ldots, x_t\}$. If a vector x_i meets the condition as in (30):

$$\frac{\max\{g_1(x_i), g_2(x_i), \ldots, g_M(x_i)\}}{\sum_{m=1}^{M} g_m(x_i)} > \gamma \tag{30}$$

we call x_i is an effective test vector. The value of γ is usually from 0.1 to 0.9, it equaled 0.3 in our experiment.

III. Incremental Learning of GMM

A. Basis of Experiments

1) Corpus: The evaluation corpus is the 1st Speech Separation Challenge (SSC) Corpus [16]. This corpus contains three sets: training, testing, and development. The training set consists of 17000 sentences (500 from each of the 34 talkers). The testing and development set consist of five different target-to-masker ratio (TMR): 6, 0, clean, -6 and -12dB. Each TMR condition contains 300 sentences in development set, 889 sentences in testing set.

2) Front-end: In order to remove the influence of environmental noise and channel noise, VAD and RASTA were used in the experiment. The central frequencies of the filters are from 200Hz to 10kHz. Then, the filtered signal of each filter is framed using 32ms frame length and 16ms frame-shift. Mel filter was composed of 48 triangle filter. The output of three lowest Mel filter was discarded to improve noise immunity.

B. Train

We trained each speaker model individually from clean training speeches in the corpus to learn each speaker characteristic distribution. If all of 500 speeches were trained, a lot of algorithms would get 100% recognition rate. So we will train the number of speeches from 20 to 100.

Train method (1): EM Method: Non-incremental training traditional EM, 20 speeches randomly selected from corpus was trained by traditional EM algorithm.

Train method (2): SOINN and EM Method: Non-incremental training by SOINN and EM, 20 speeches randomly selected from corpus was trained by SOINN and EM algorithm.

Train method (3): SOINN and EM incremental Method: Incremental training, 20 speeches randomly selected from corpus was trained by SOINN and IEM algorithm, and 80 speeches randomly selected from the rest 480 speeches would be trained four times in incremental learning (20 speeches once a time).

C. Test

Test method (1): Test but not incremental learning.

Test method (2): Test and incremental learning, the test speeches would be trained according to the test result.

D. Experiment Results and Analysis

We evaluated our algorithm with above clean speech subset of SSC corpus.

TABLE I
THE RESULT OF THE PROPOSED ALGORITHM

Train	Test	Result
EM	Test	97.50%
EM	Test&Learn	98.75%
SOINN-EM	Test	98.50%
SOINN-EM	Test&Learn	100.0%
SOINN-iEM	Test	100.0%
SOINN-iEM	Test&Learn	100.0%

Table 1 shows that SOINN-EM algorithm done better than traditional EM algorithm. This is due to SOINN could adaptively give the clusters number of each speaker model. Table 1 also shows that SOINN-IEM is better than traditional EM. The reason is that IEM algorithm improved the speaker model constantly in incremental learning while the number of initial sample is not rich.

E. Evaluation Different Configure Parameters

Here, we will show the experiment evaluations in different configure parameters.

1) The effect of the SOINN: We examined the effect of SOINN by comparing with K-means in initialization of GMM. K-means method has the parameter K that needs to be determined in advance. This parameter K is also the number of mixture component of GMM. So we test different K values compared with SOINN. Table 2 shows that the best result of K-means is 99.33% and K=100, it is also poor than SOINN.

2) The effect of parameter λ: We also examined the effect of parameter λ in formula (12) in development set, as shown in table 3, the chosen results with λ=3.5 and λ=5.5 show much higher performance compared with the results without limitation of with λ=infinity.

TABLE II

THE RESULT OF SOINN AND K-MEANS

Method	Test	Result
K-means	K=40	97.85%
K-means	K=50	98.21%
K-means	K=60	98.45%
K-means	K=80	98.87%
K-means	K=100	99.33%
K-means	K=150	98.80%
SOINN	—	100.0%

TABLE III

THE RESULT WITH DIFFERENT Λ IN GMM

λ	EM	SOINN-EM
3	98.65%	99.66%
3.5	99.80%	100.0%
4	99.75%	99.77%
4.5	96.28%	99.89%
5	96.40%	99.89%
5.5	99.85%	100.0%
6	99.89%	99.89%
6.5	96.40%	99.89%
7	99.89%	99.89%
infinity	99.75%	99.89%

3) EM and SOINN-EM: The effect of different amount of training speeches with EM and SOINN-EM has been shown in table IV.

TABLE IV

THE RESULT WITH DIFFERENT TRAINED SPEECHES

Trained Speeches	EM	SOINN-EM
5s	97.50%	97.85%
10s	98.98%	99.54%
15s	99.32%	99.88%
20s	99.85%	100.0%
30s	100.0%	100.0%

IV. CONCLUSIONS

In this paper, theories of incremental learning and self-adaption are applied to propose an incremental learning GMM algorithm. The neural network SOINN was combined to GMM model to take the place of K-means algorithm. With SOINN, we can easily find out how many clusters a speaker model consist of. With this incremental learning method, huge amounts of data could be processed gradually and incrementally. So, characteristics which were not studied in initialization or changing with time will be learned in incremental learning. Of course, there are still many aspects to be improved, for example, the better combination of SOINN and GMM, the usage of full covariance matrices.

REFERENCES

[1] Reynolds, A. Douglas, Quatieri, F. Thomas, and Dunn, B. Robert, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*,vol. 10, pp. 19-41, Jan. 2000.

[2] D. A. Reynolds, R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker model," *IEEE Trans. Speech Audio Process.*, vol.3, pp. 72-83, Jan. 1995.

[3] T. G. Clarkson, C. C. Christodoulou, Y. Guan, D. Gorse, D.A. Romano-Critchley, and J. G. Taylor, "Speaker identification for security systems using reinforcement trained pRAM neural network architectures," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 31, no. l, pp. 65-76, Feb. 2001.

[4] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Society*, vol. 13, no. 5, pp. 308-311, May. 2006.

[5] B. Xiang and T. Berger, "Efficient text-independent speaker verification with structural Gaussian mixture models and neural network," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp.447-456, Sept. 2003.

[6] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transaction on Speech and Audio Processing*, vol. 2, no. 2, pp. 291-298, April 1994.

[7] R. Jourani, K. Daoudi, R. Andr-Obrecht, and D. Aboutajdine, "Large Margin Gaussian mixture models for speaker identification," *in Proc. of Interspeech*, vol. 1, pp. 1441-1444, Nov. 2010.

[8] Hikaridai, Seika-cho and Soraku-gun, "Split and merge EM algorithm for improving Gaussian mixture density estimates," *Journal of VLSI Signal Processing*, vol. 26, pp. 133-140, Aug. 2000.

[9] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Journal of Computer Speech and Language*, vol. 9, pp. 171-185, Apr. 1995.

[10] Furao Shen and Osamu Hasegawa, "Self-organizing incremental neural network and its application", *International Conference on Artificial Neural Network*, vol. 6354, pp. 535-540, Sept. 2010.

[11] D. A. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. on SAP*, vol.3, no. 1, pp. 72-83, Jan. 1995.

[12] S. B. Davies and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357-366, Aug. 1980.

[13] G. S. Ying, C D. Mitchell and L H. Jamieson, "Endpoint detection of isolated utterances based on a modified teager energy measurement," *Acoustics, Speech, and Signal Processing*, vol. 2, pp. 732-735, Apr. 1993.

[14] J. G. Wilpon , B. H. Juang , L. R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, vol. 12, pp. 821-824, Apr. 1987

[15] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, Dec. 1977.

[16] M. P. Cooke, T. Lee, 2006. Speech separation challenge, http://www.dcs.shef.ac.uk/ martin/speechseparationchallenge.htm

[17] Youki Kamiya, Shen Furao and Osamu Hasegawa, "A Self-organized Incremental Network for Online Supervised Learning and Topology Learning," *Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems*, CD-ROM FR-H4-3, Tokyo, Japan, Sep. 2006.

[18] Shen Furao, Tomotaka Ogura, Osamu Hasgeawa, "An enhanced self-organizing incremental neural network for online unsupervised learning," *Neural Networks*, vol.20, pp.893-903, July. 2007.

[19] Shen Furao and Osamu Hasegawa, "A Fast Nearest Neighbor Classifier Based on Self-organizing Incremental Neural Network," *Neural Networks*, vol. 21, pp. 1537-1547, July. 2008.