Probabilistic Self-Organizing Maps for Text-Independent Speaker Identification

Ayoub Bouziane*[1], Jamal Kharroubi[2], Arsalane Zarghili[3]
Intelligent Systems and Applications Laboratory, Sidi Mohamed Ben Abdellah University
P.B: 2202 Immouzer road, Fez, Morocco
*Corresponding author, e-mail: ayoub.bouziane@usmba.ac.ma[1], jamal.kharroubi@usmba.ac.ma[2], arsalane.zarghili@usmba.ac.ma[3]

Abstract

The present paper introduces a novel speaker modeling technique for text-independent speaker identification using probabilistic self-organizing maps (PbSOMs). The basic motivation behind the introduced technique was to combine the self-organizing quality of the self-organizing maps and generative power of Gaussian mixture models. Experimental results show that the introduced modeling technique using probabilistic self-organizing maps significantly outperforms the traditional technique using the classical GMMs and the EM algorithm or its deterministic variant. More precisely, a relative accuracy improvement of roughly 39% has been gained, as well as, a much less sensitivity to the model-parameters initialization has been exhibited by using the introduced speaker modeling technique using probabilistic self-organizing maps.

Keywords*: speaker identification system, gaussian mixture model (GMM), probabilistic self-organizing maps, EM algorithm, deterministic annealing EM algorithm, the SOEM algorithm.*

1. Introduction

The Gaussian mixture models (GMMs) [1], [2] are considered as the simplest and the traditional speaker modeling technique in speaker recognition systems, as well as, the basis of the most successful approaches that have been emerged in the last decade.

Each speaker is modeled in the system as a mixture of Gaussian densities, which may reflect the specific acoustical classes of the speaker. Generally, the parameters of the Gaussian mixture models (GMMs) are estimated using the widely used and well-known EM algorithm. Beside all the advantages of the EM algorithm, such as its simplicity, both conceptually and computationally, it suffers from some general drawbacks like its sensitivity to the initial model parameters - especially in a multivariate context - and the trapping in local optimums. To overcome this problem, various techniques were proposed and used in the speaker recognition state-of-the-art, such as the deterministic annealing EM proposed by Ueda and Nakano [3],the split and merge algorithms, as well as some heuristics to find the appropriate initial points for the EM algorithm.

In the same perspective, the probabilistic self-organizing maps method [4]–[6], based on the combination between the strengths of self-organizing maps and mixture models, was proposed and yielded better results in some image processing applications. In the present study, the probabilistic self-organizing maps method is assessed and introduced for speaker modeling in speaker recognition applications. The obtained results using the probabilistic self-organizing maps are compared with the classical training of the Gaussian mixture models using the EM algorithm and its deterministic variant.

The remainder of this paper is organized as follows. The second section briefly highlights the general operating structure of speaker identification systems. Section 3 and 4 deal with speaker modeling process, Section 3 gives a brief description of Gaussian Mixture Models and outlines the principle of the EM algorithm and its deterministic annealing variant, while section 4 introduces the Probabilistic Self-Organizing Maps for speaker modeling in speaker recognition systems. Next, the experimental results are provided in Section 5. Finally, conclusions and future directions are drawn in Section 6.

Received Mar 5, 2017; Revised September 18, 2017; Accepted September 30, 2017

2. The General Operating Structure of the Speaker Identification Systems

The basic structure of automatic speaker identification systems, as shown in Figure 1, consists of two distinct phases: the training phase and the testing phase.

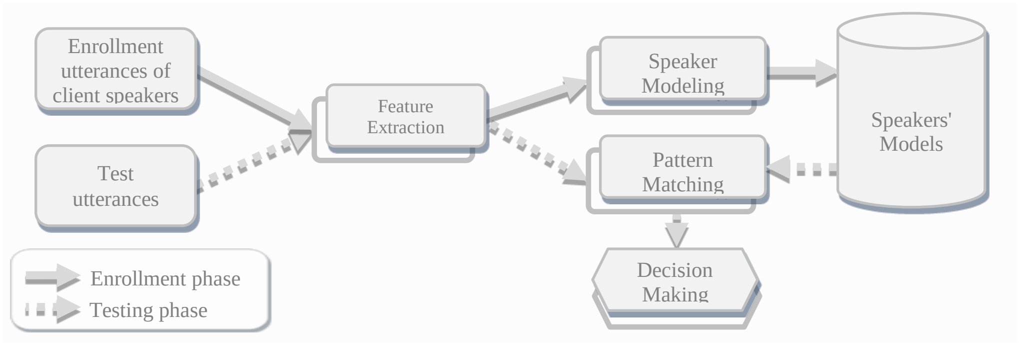

Figure 1. The basic framework and components of speaker recognition systems

During the training phase, speech samples are gathered from new client speakers, their individual feature vectors that reflect the characteristics of their vocal tracts are extracted and they're used to train a reference model for each client speaker. As regards the testing phase, the speech signal of the unknown speaker is acquired, corresponding feature vectors are extracted and scored against the previously enrolled reference models. Finally, the similarity scores computed from this comparison are then used to make a decision about the identity of the speaker.

3. Speaker Modeling Using the Traditional Gaussian Mixture Models

The Gaussian Mixture Models were firstly introduced to the speaker recognition community in 1995 [7]–[9]. Since then, they have become the predominant approach for speaker modeling in text-independent speaker recognition systems, and the basis of the most successful approaches that have emerged in the last decade. The basic idea underlying the GMM approach consists in modeling the distribution of the speaker’s features as a Gaussian mixture density. The Gaussian mixture density is generally defined by a weighted sum of M Gaussian densities, as depicted in Figure 2, and is given by the following equation:

$$p(x_t|\lambda) = \sum_{i=1}^{M} w_i b_i(x_t) = \sum_{i=1}^{M} w_i\, g(x_t|\mu_i, \Sigma_i) \quad (1)$$

where, x_t is a D-dimensional feature vector, b_i (x)=g(x│μ_i,Σ_i),i=1,2,3,…,M. are the Gaussian densities and 〖 w〗_i,i=1,2,3,…,M are the mixture weights. Each density component is a D-variate Gaussian function of the following form:

$$g(x_t|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x_t - \mu_i)^T \Sigma_i^{-1}(x_t - \mu_i)\right\} \quad (2)$$

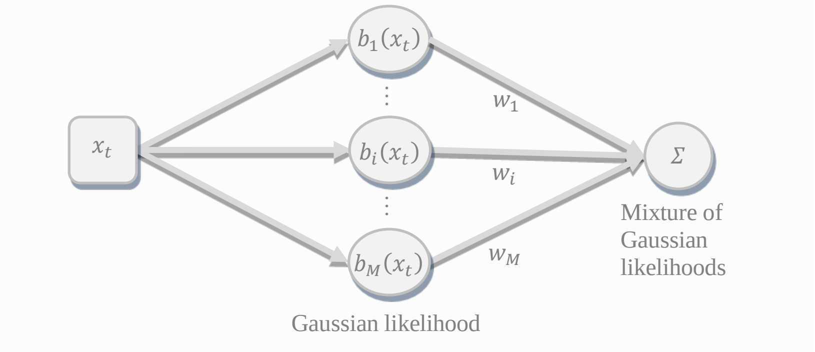

Figure 2. Gaussian Mixture Density

The Gaussian mixture model λ is parameterized by the collection of the mean vectors, covariance matrices and mixture weights of the Gaussian densities λ={w_i,μ_i,Σ_i },i=1,2,…,M. The mixture weights, w_i, furthermore satisfy the constraint ∑_(i=1)^M▒w_i =1.

The motivation behind the use of Gaussian mixture models for speaker modeling lies on the assumption that Gaussian densities may model a set of hidden acoustic classes that reflect the characteristics of the speaker dependent vocal tract.

The model parameters λ={w_i,μ_i,Σ_i },i=1,2,…,M. are determined in such manner that they best fit the distribution of the training feature vectors X={x1,. . . ,xT}. In other words, they are determined in such manner that they maximize the log-likelihood of the GMM log(p(X│λ)).The traditional and the commonly-used method in this context is the maximum likelihood estimation (MLE) method via the Expectation–maximization (EM) algorithm.

3.1. Gaussian mixture models using the EM algorithm

The basic idea of the EM algorithm, as reported in algorithm 1, consists in starting with an initial model λ and tending to estimate a new model(λ)¯, such that: p(X│λ)≥p(X│λ¯).Next, the new estimated model λ¯ becomes an initial model to be refined in the next iteration, and the process is repeated until an increase in the log-likelihood of the data, given the current model, is less than some convergence threshold.

Algorithm 1. The EM algorithm

Input : Training feature vectors $\mathrm{X} = \{\mathrm{x}_1, . . . , \mathrm{x}_1\}$

Output : GMM of M component $\{\mu_c, \sigma_c, w_c\}_{c=1}^{M}$.

1: Randomly initialize the model parameters $\{\mu_c, \sigma_c, w_c\}_{c=1}^{M}$.

2: Compute the a posteriori probability $P(i|x_t, \lambda)$:

$$P(i|x_t,\lambda) = \frac{w_i g(x_t|\mu_i, {\sigma^2}_i)}{\sum_{k=1}^{M} w_k g(x_t|\mu_k, {\sigma^2}_k)} \quad (1)$$

3: Re-estimate the new model parameters, i.e. the mixture weights, the means and variances vectors, using the following equations:

$$w_i = \frac{1}{T}\sum_{t=1}^{T} P(i|x_t,\lambda) \qquad \mu_i = \frac{\sum_{t=1}^{T} P(i|x_t,\lambda)x_t}{\sum_{t=1}^{T} P(i|x_t,\lambda)} \qquad {\sigma^2}_i = \frac{\sum_{t=1}^{T} P(i|x_t,\lambda){x_t}^2}{\sum_{t=1}^{T} P(i|x_t,\lambda)} - {\mu_i}^2 \quad (2)$$

5: Repeat step 2-3 until convergence.

6: Return the model parameters $\{\mu_c, \sigma_c, w_c\}_{c=1}^{M}$.

3.2. Gaussian mixture models uisng the DAEM algorithm

The Deterministic Annealing EM algorithm [3] is an EM variant algorithm based on the deterministic annealing concept. The key idea of the DAEM algorithm consists in reformulating the problem of maximizing the log-likelihood in the classical EM algorithm as a problem of minimizing the thermodynamic free energy defined through the maximum entropy principle and statistical mechanics analogy.

Similarly to the EM algorithm, the DAEM algorithm is an iterative procedure based on expectation and maximization steps. In the expectation step, a new temperature-parameterized posterior distribution was introduced as follows:

$$P_\beta(i|x_t,\lambda) = \frac{w_i{}^\beta g(x_t|\mu_i,\Sigma_i)^\beta}{\sum_{k=1}^{M} w_k{}^\beta g(x_t|\mu_k,\Sigma_k)^\beta} \quad (5)$$

where the temperature 1/β is gradually decreased during the training process, and the posterior distribution is optimized at each temperature. The diminishing rate of temperature must be as slow as possible, particularly at the early stages of training. In the maximization step, the model parameters are estimated using the temperature-parameterized posterior distribution P_β (i│x_t,λ) in exactly the same way as the classical EM algorithm. See Figure 3.

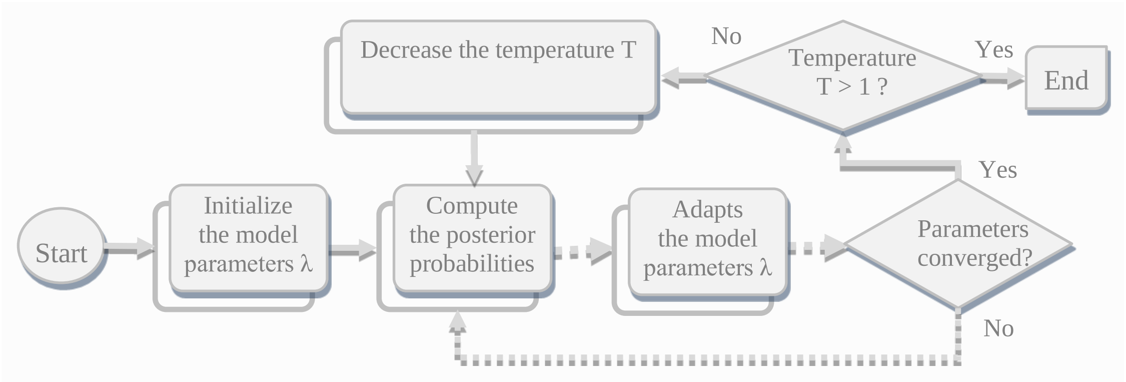
Decrease the temperature T
No
Temperature T > 1 ?
Yes
End
Yes
Start
Initialize the model parameters λ
Compute the posterior probabilities
Adapts the model parameters λ
Parameters converged?
No

Figure 3. Flowchart of the DAEM algorithm

3. Speaker modeling using Probabilistic Self-Organizing Maps (PbSOM)

The Self-Organizing Maps (SOM), commonly known also as Kohonen network [10], are the most popular unsupervised neural network for data clustering and visualization. The SOM approach was inspired from self-organizing nature of the human cerebral cortex. Indeed, it is based on the idea of competition and neighborhood update concepts which preserve the topological relationships between classes in the network [11].

A self-organizing map, as shown in Fig 4, consists of two layers of neurons, an input layer and an output layer. The input layer is composed of N input neurons according to the N input vectors {X_n=[X_1 ,X_2,…,X_N],1≤n≤N} to be classified, while the output layer (so-called competitive layer) is composed of M output neurons {r_m=[r_1 ,r_2,…,r_M],1≤m≤M} according to the M clusters {C_m=[C_1 ,C_2,…,C_M],1≤m≤M} to be determined. The input neurons are fully connected to output neurons, which are connected to each other by a neighborhood relation h_ij,1≤i,j≤M dictating the structure of the layer. The layer structure is often specified by the following factors: the local lattice structure (hexagonal, rectangular …) and the dimension or the global map shape (sheet, cylinder …). The self-organizing map algorithms are trained iteratively based on two steps: a competitive step and a cooperative step. In the first step, the various output neurons compete with each other to determine the “winner” neuron(s) which best matches the input vector(s). In the second step, i.e. the cooperative step, the weights of the winner neuron(s) and that of neurons close to them in the SOM lattice are adjusted towards the input vector(s). Therefore, output neurons will self-organize to an ordered map in such a way that output neurons which have similar weights will be placed nearby after training.

Once the original SOM idea been proposed and succeeded in several clustering applications, a numerous variations and improvements of the original idea have been proposed in the literature. Among the proposed ones are the probabilistic self-organizing maps.

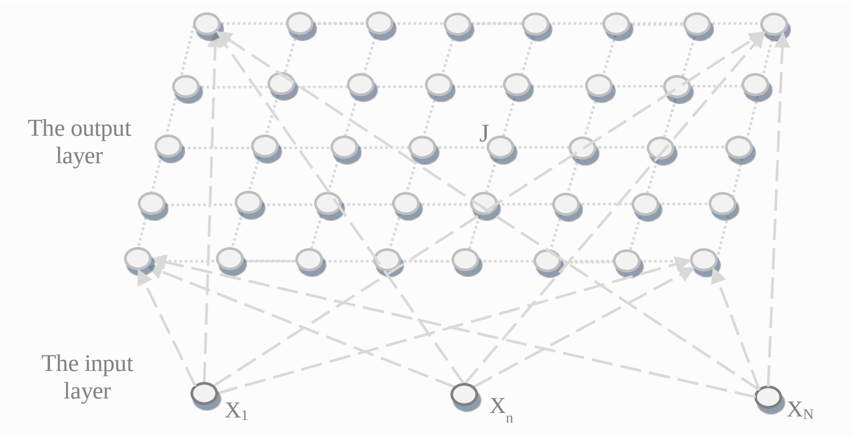

Figure 4. Structure of Self-Organizing Map

The Probabilistic Self-Organizing Maps are a probabilistic variant of the traditional self-organizing maps where the response n_k of each neuron θ_k to each input vector x_i is modeled by a multivariate Gaussian〖 θ〗_k={w_k,μ_k,Σ_k }, as follows:

$$n_k(x_i;\theta_k)=\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}}\exp(-\frac{1}{2}(x_i-\mu_k)^T|\Sigma_k|^{-1}(x_i-\mu_k)) \qquad (6)$$

In the literature, several formulations and algorithms have been proposed for the training of the probabilistic self-organizing map. Among the most widely studied and applied ones is the coupling-likelihood mixture model formulation together with the SOEM algorithm [4], [5].

The coupling-likelihood mixture model formulation was principally inspired from the work of Sum and john that interpreted Kohonen’s sequential SOM learning algorithm as maximizing the local correlations (coupling energies) between the output neurons and their neighborhoods with the input traning data [12].

Given a SOM Network ℵ of M output neurons where each neuron n_(k)is parameterized by a reference Gaussian〖 θ〗_k={w_k,μ_k,Σ_k } . The coupling energy between of each neuron n_k and its neighborhood in terms of probabilistic likelihood is defined as follows [5]:

$$p_s(x_i|k;\lambda,h)=n_k(x_i;\theta_k)^{h_{kk}}\prod_{l\neq k} n_l(x_i;\theta_l)^{h_{kl}} \qquad (7)$$

Here, λ={θ_l,θ_2,…,θ_M } is the reference model of the whole SOM Network ℵ, h_kl denotes the neighborhood function that defines the strengths of lateral interaction between neurons k and l ∈{1,2,…,M} and the term ∏_(l≠k)▒〖n_l (x_i;θ_l)〗^kl represents the neighborhood response of the neuron〖 n〗_k. Accordingly, the coupling likelihood (the coupling energy) of an input data x_i over the network ℵ can be depicted as shown in Fig. 5 and defined by the following mixture likelihood:

$$p_s(x_i;\lambda,h)=\sum_{k=1}^{M}w_s(k)p_s(x_i|k;\lambda,h) \quad (8)$$

Compared to the GMM traditional formulation, the coupling-likelihood mixture model formulation embeds a coupling-likelihood layer between the Gaussian-likelihood layer and the mixture-likelihood layer in order to take into account the coupling between the neurons and their neighborhoods, see Fig 5.

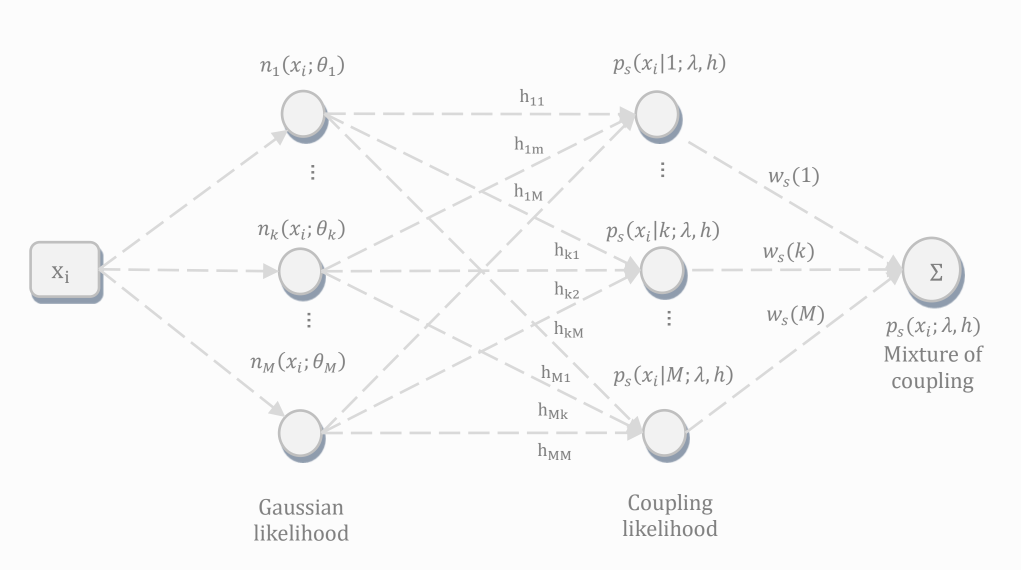

Figure 5. The coupling-likelihood x_i over the network

Algorithm 2. The SOEM algorithm

Input : Training feature vectors $X=\{x_1,...,x_1\}$

Output : Optimized Gaussian mixture Model parameters.

1: Randomly initialize the model parameters $\lambda=\{\theta_l\}_{l=1}^{M}=\{\mu_l,\sigma_l,w_l\}_{l=1}^{M}$.

2: Initialize the radius of the neighborhood function at a higher value.

3: Repeat the following steps until convergence:

❖ Expectation step: Aims to compute the posterior probability of the Gaussian components representing the network neurons for each x_i:

$$\gamma_{(k|i)}=p_s(k|x_i;\lambda,h)=\frac{\exp\left(\sum_{l=1}^{M}h_{kl}\log\left(n_l(x_t;\theta_l)\right)\right)}{\sum_{j=1}^{M}\exp\left(\sum_{l=1}^{M}h_{jl}\log\left(n_l(x_t;\theta_l)\right)\right)} \quad (3)$$

❖ Maximization step: Aims to re-estimation of the networks parameters i.e. the mean and variances vectors, using the following equations:

$$\mu_l=\frac{\sum_{i=1}^{N}(\sum_{k=1}^{M}\gamma_{(k|i)}h_{kl})x_t}{\sum_{i=1}^{N}(\sum_{k=1}^{M}\gamma_{(k|i)}h_{kl})} \quad (4)$$

$$\Sigma_l=\frac{\sum_{i=1}^{N}(\sum_{k=1}^{M}\gamma_{(k|i)}h_{kl})(x_t-\mu_l)(x_t-\mu_l)^T}{\sum_{i=1}^{N}(\sum_{k=1}^{M}\gamma_{(k|i)}h_{kl})} \quad (5)$$

```
4:  Decrease the radius of the neighborhood function.
5:  Repeat step 3-4 until it reaches a predefined minimum value.
6:  Return the model parameters $\{\mu_l, \sigma_l, w_l\}_{l=1}^{M}$.
```

The neighborhood function is traditionally taken as a Gaussian kernel of the following form:

$$h_{kl} = \exp\left(-\frac{\|r_k - r_l\|}{2\sigma^2}\right) \quad (12)$$

where ǁr_k-r_l ǁ is the Euclidean distance between two neurons r_kand r_l, and σ is the radius of the neighborhood function. On another side, the network parameters, i.e., the reference model λ, are determined using the SOEM algorithm aiming to maximize the following objective log-likelihood function:

$$L_s(\lambda;\chi,h) = \log\left(\prod_{t=1}^{N} p_s(x_i;\lambda,h)\right) = \sum_{t=1}^{N} \log\big(p_s(x_i;\lambda,h)\big) \quad (12)$$

The SOEM algorithm is a modified EM algorithm that iteratively refines the network parameters by alternating between modified expectation and maximization steps, until convergence. The specifics of the SOEM algorithm are reported in Algorithm 2 and depicted as flowchart in Figure 6.

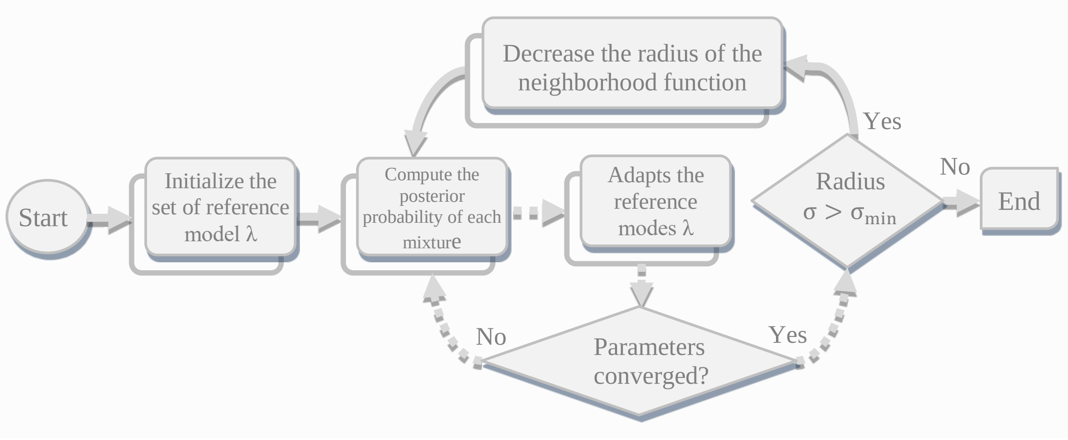
Decrease the radius of the neighborhood function
Yes
Start
Initialize the set of reference model λ
Compute the posterior probability of each mixture
Adapts the reference modes λ
Radius σ > σ_min
No
End
No
Parameters converged?
Yes

Figure 6. Flowchart of the SOEM algorithm

4. Experiments, Results and Discussion

The aim of the performed experiences in this study is to access and evaluate the performance of introduced speaker modeling technique using probabilistic self-organizing maps compared to the traditional technique using the EM algorithm or its deterministic variant.

5.1. Experimental Protocol

The conducted experiments in this study performed under a speech corpus of 40 Moroccan speakers in the age range of 18 to 30 years, 17 female and 23 male. Each speaker was recorded for at least more than two recording sessions separated by around two-three weeks. The sort of recorded speech incorporates free monolog in Moroccan dialect and read text in Arabic, French and English languages. The recordings were gathered from volunteer speakers over internet as voice messages via Skype. In order to cover a wide range of real-life acoustical environments, we recommended the speakers to make calls from many different places, e.g., home, office etc. Furthermore, different kinds of equipment were used for recoding (laptops, tablets and smartphones …). On another side, the voice messages were digitized at

16 kHz with a determination of 16 bits (mono, PCM) and stored in the most commonly used “wav” format.

The feature vectors of the speakers’ speech utterances were extracted using the mel-frequency cepstral coefficients [13]. Each frame was parameterized by a vector consisting of 19 coefficients. The MFCCs features are pre-processed as follows. The emphasizing step is firstly performed using a simple first order filter with transfer function: H (z) = 1 – 0.95z. Next, the emphasized speech signal is blocked into Hamming-windowed frames of 25 ms (400 samples) in length with 10 ms (160 samples) overlap between any two adjacent frames [13].

During the training phase, one minute of active speech per speaker is used for the building the speaker’s model, whereas in the testing phase, the evaluation data composes 400 identification tests of 8 seconds (i.e., ten tests per speaker each of 8s in duration).

On another side, the temperature β of the DAEM algorithm was updated using following way: β(i)= √(i/I),i = 1,2,...,I. where β(i) is the value of β at i-th temperature update step, and I is the total number of temperature update steps (Empirically chosen as I=10). Regarding the SOEM algorithm, the probabilistic self-organizing maps were trained on rectangular lattices using the Gaussian kernel h_kl as neighborhood function. The neighborhood width is fixed in the beginning at σ=1 and reduced gradually during the training to 0.

5.2. Sub Bab 2

The identification performances of the introduced PbSOM-based modeling technique and the traditional GMM-based modeling techniques using the EM and the DAEM algorithms are summarized in Figure 7. As it can be seen, the performance evaluation was done at various models’ sizes (i.e. number of Gaussian components used for speaker modeling). Moreover, the experiments were repeated three times using the same experimental protocol and the same model size in order to evaluate the techniques’ sensitivity to the initial parameters.

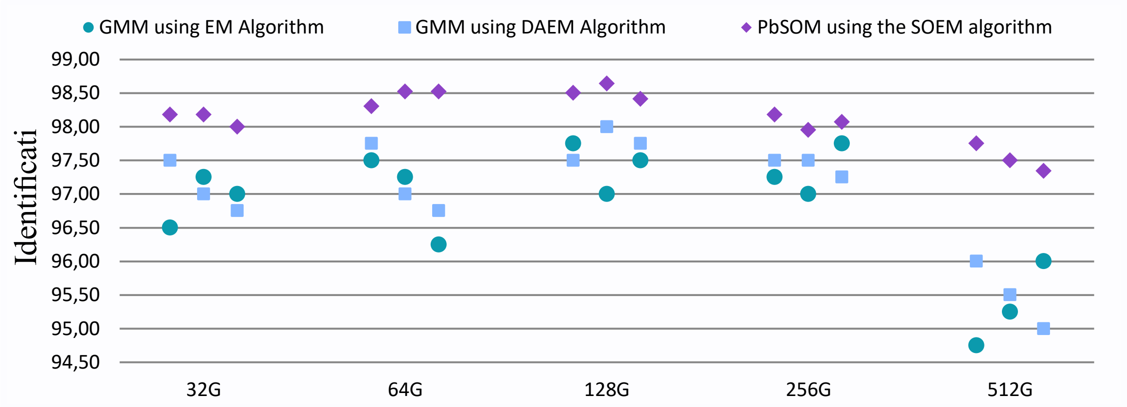

Figure 7. The performance of the introduced PbSOM-based modeling technique using the SOEM algorithm compared to the traditional GMM-based modeling techniques using the EM and the DAEM algorithms

The obtained results clearly confirm the superiority of the introduced technique using the SOEM algorithm in comparison with the traditional technique using the classical GMMs and the EM algorithm or its deterministic variant. Effectively, it can be seen across the various used model sizes that the DAEM algorithm outperforms the EM algorithm and the SOEM algorithm significantly outperforms both EM and DAEM algorithms. By way of illustration, we can see that the identification performance of the DAEM–based system using models’ size of 128 Gaussians demonstrates a relative accuracy improvement of roughly 11% compared to the system performance using the EM algorithm. Likewise, we can observe that the identification performance of the SOEM–based system using the same models’ size (i.e. 128) demonstrates a relative accuracy improvement of approximately 39% and 32% compared to the system performances using the EM and the DAEM algorithms, respectively.

Concerning the algorithms sensitivity to the parameters initialization, we can observe that the system performance using EM algorithm is severely unstable when repeating the same experiment using the same model size and experimental protocol. Apparently, the EM algorithm seems to be strongly dependent on the model-parameters initialization. Besides, we can remark that the DAEM algorithm is less sensitive to the parameters initialization compared to the EM algorithm. On another hand, we can see that the SOEM algorithm is much less sensitive to the parameters initialization compared to the EM and the DAEM algorithms. Seemingly, the self-organizing quality of the SOEM algorithm makes it less sensitive to parameters initialization.

6. Conclusion

In this paper, a novel speaker modelling technique using the probabilistic self-organizing maps (PbSOMs) has been introduced for text-independent speaker identification. The basic motivation behind the introduced technique was to combine the strengths of the traditional self-organizing maps and the Gaussian mixture models. Experimental results demonstrated that the introduced modelling technique using probabilistic self-organizing maps outperforms the traditional technique using the classical GMMs and the EM algorithm or its deterministic variant.

References

[1] D. Reynolds, « Gaussian Mixture Models », in *Encyclopedia of Biometrics*, S. Z. Li et A. K. Jain, Éd. Boston, MA: Springer US, 2015, p. 827‑832.

[2] T. R. J. Kumari et H. S. Jayanna, « Limited Data Speaker Verification: Fusion of Features », *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, n^o 6, p. 3344‑3357, déc. 2017.

[3] N. Ueda et R. Nakano, « Deterministic annealing EM algorithm », *Neural Networks*, vol. 11, n^o 2, p. 271‑282, mars 1998.

[4] S.-S. Cheng, H.-C. Fu, et H. Wang, « CEM, EM, and DAEM Algorithms for Learning Self-Organizing Maps », in *2007 IEEE Workshop on Machine Learning for Signal Processing*, 2007, p. 378‑383.

[5] S.-S. Cheng, H.-C. Fu, et H. Wang, « Model-Based Clustering by Probabilistic Self-Organizing Maps », *IEEE Transactions on Neural Networks*, vol. 20, n^o 5, p. 805‑826, mai 2009.

[6] L. J. Lin Chang, « Skin detection using a modified Self-Organizing Mixture Network », p. 1‑6, 2013.

[7] D. A. Reynolds et R. C. Rose, « Robust text-independent speaker identification using Gaussian mixture speaker models », *IEEE Transactions on Speech and Audio Processing*, vol. 3, n^o 1, p. 72‑83, janv. 1995.

[8] D. A. Reynolds, « Automatic speaker recognition using gaussian mixture speaker models », *The Lincoln Laboratory Journal*, p. 173–192, 1995.

[9] D. A. Reynolds, « Speaker identification and verification using Gaussian mixture speaker models », *Speech Communication*, vol. 17, n^o 1, p. 91‑108, août 1995.

[10] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.

[11] T. Heskes, « Self-organizing Maps, Vector Quantization, and Mixture Modeling », *Trans. Neur. Netw.*, vol. 12, n^o 6, p. 1299–1305, nov. 2001.

[12] J. Sum, C. Leung, L. Chan, et L. Xu, « Yet Another Algorithm Which Can Generate Topography Map », *IEEE Trans. Neural Networks*, vol. 8, p. 1204–1207, 1997.

[13] B. Ayoub, K. Jamal, et Z. Arsalane, « An analysis and comparative evaluation of MFCC variants for speaker identification over VoIP networks », in *2015 World Congress on Information Technology and Computer Applications Congress (WCITCA)*, 2015, p. 1‑6.