Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 15 (2): 295-300, 2015
ISSN 1812-5654

ansinet
Asian Network for Scientific Information

RESEARCH ARTICLE OPEN ACCESS

DOI: 10.3923/jas.2015.295.300

FLVQ Based GMM in Speaker Verification

P. Shanmugapriya and Y. Venkataramani
Department of Electronics and Communication Engineering, Saranathan College of Engineering, Venkateswara Nagar, Panjapur, Trichy, 620012, Tamil Nadu, India

ARTICLE INFO
Article History:
Received: August 03, 2014
Accepted: November 08, 2014

Corresponding Author:
P. Shanmugapriya,
Department of Electronics and Communication Engineering,
Saranathan College of Engineering,
Venkateswara Nagar, Panjapur, Trichy,
620012, Tamil Nadu, India
Tel: 9994539389

ABSTRACT
In order to improve the verification rate of automatic speaker verification system, a novel training algorithm for Gaussian Mixture Model is proposed in this study. A novel feature extraction method for automatic speaker verification system is also presented. This system includes extraction of discrete wavelet transform based Mel frequency cepstral coefficients from speech and Fuzzy Learning Vector Quantization based gaussian mixture model training. This feature extraction approach utilizes the dynamic spectral features which are useful for recognizing the speaker. The proposed training method for speaker model not only reduces the number of features vectors used to train the model but also increases the verification rate than the conventional GMM-expectation maximization algorithm. The proposed method of speaker verification is evaluated using TIMIT database. Experiments are also conducted with other vector quantization algorithms: (1) Learning vector quantization, (2) K-means, (3) Fuzzy C-means and (4) Linde-buzo-grey algorithm as training algorithms for GMM. Experimental results demonstrate that the performance of the proposed system is better when compared to conventional systems in terms of verification rate.

Key words: Automatic speaker verification system, Gaussian mixture model, fuzzy learning vector quantization, DWT-MFCC features

INTRODUCTION

The main objective of vector quantization is to compress a large number of short term spectral vectors into a small set of code vectors. The set of code vectors is named as codebook. Each code vector represents the short-term spectral variations in the speech due to different textual content (He *et al*., 1997). The codebook is designed by means of clustering algorithms such as K-means (Gray, 1984), possibilistic C-means (Kummamuru *et al*., 2003) and fuzzy C-means (Bezdek, 1981). The VQ based system modeling provides high recognition accuracy (Equitz, 1989) with reduced set of feature vectors. A VQ codebook can also be trained with LBG algorithm (Linde *et al*., 1980) which minimizes the quantization error. Here, the codebook vectors selected based on the minimum distance criterion represents the distribution of feature vectors. But the demerit of LBG algorithm based codebook training is its weak discriminative power due to the fact that only the samples within a class but no competitive data, have been used during the training process. Another approach has been proposed by Kohonen (1990) to globally optimize the codebooks with a certain unsupervised learning algorithm. These algorithms are called Learning Vector Quantization (LVQ). Instead of finding the mean vector of a cluster which approximates distribution of training samples, the codebooks trained with LVQ algorithms define directly the classification borders between classes according to the nearest-neighbor rule. However, the classification decision for a speaker depends on the vector sequence derived from a test sentence rather than on an individual vector, thus, a higher correct classification rate for feature vectors achieved with the LVQ codebooks does not necessarily lead to a higher speaker identification rate. This is because the feature vectors are highly correlated and this correlation has not been taken into consideration in the LVQ algorithm. In order to introduce the correlation between

feature vectors in the training algorithm and to utilize the merits of LVQ, fuzzy LVQ algorithm is used to determine the optimal codebook. Batch fuzzy LVQ algorithms were introduced by Tsao *et al.* (1994).

The fuzzy logic methods which can be efficiently used in pattern recognition (Ho *et al.*, 2001), are mainly based on fuzzy clustering analysis. The most representative fuzzy clustering algorithm is the fuzzy c-means method (Bezdek *et al.*, 1984). A reliable implementation of the fuzzy c-means should be based on assigning each training vector to the cluster center that has the maximum membership degree. But such a crisp interpretation of fuzzy c-means may exhibit serious effects on the quality of the final codebook. There are two general frameworks to solve this problem. The first one is based on Fuzzy Vector Quantization (FVQ), where special strategies for the smooth transition from the fuzzy to crisp mode have been developed. Such kind of approach was proposed by Karayiannis and Pai (1996). The second framework is the fuzzy LVQ (FLVQ) algorithm introduced by (Tsao *et al.*, 1994). In this case, the transition from fuzzy to crisp mode is accomplished by manipulating the fuzziness parameter of the fuzzy c-means. The integration of this model with the fuzzy c-means algorithm was established by Karayiannis and Bezdek (1997). A major difference between these two frameworks is that the FVQ keeps the fuzziness parameter constant throughout the training process while the FLVQ manipulates it.

This study proposes a method of modeling the speaker using the state of art GMM which is trained through FLVQ algorithm. Speaker modeling with FLVQ is proposed, experimented and contrasted it with the popular EM algorithm training process. In conventional approach, the GMM parameters, mean, covariance and weights are updated by expectation maximization algorithm. But, in this approach, the parameters are updated through FLVQ algorithm. This approach reduces the computational complexity and improves the performance of speaker verification system. The training of GMM through this method also provides a lower classification error rate.

MATERIALS AND METHODS

Speaker verification system: Speaker recognition is the process of finding a speaker from his/her speech. This system works in two main phases. One is the training phase or enrolment phase which creates a model for the vocal characteristics of speaker. It involves feature extraction and model generation. The second phase is the verification phase in which a decision is made according to the score obtained by the speaker. It involves the comparison and decision making. The block diagram of the proposed system with DWT based MFCC features used for GMM trained with FLVQ is shown in Fig. 1.

Feature extraction: In speech signal processing, MFCC is a representation of the short-term power spectrum of a sound, based on a linear discrete cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. In MFCC, the frequency bands are equally spaced on the Mel scale which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound. In this study, the wavelet transformed MFCC coefficients are obtained by decomposing the speech signal into different resolution levels and then MFCCs are extracted from the wavelet channels. This represents the frequency characteristics of speech signal at different resolution levels (Zhang and Benveniste, 1992). Thus the combination of wavelet transform and MFCC can represent sound signals in an efficient way.

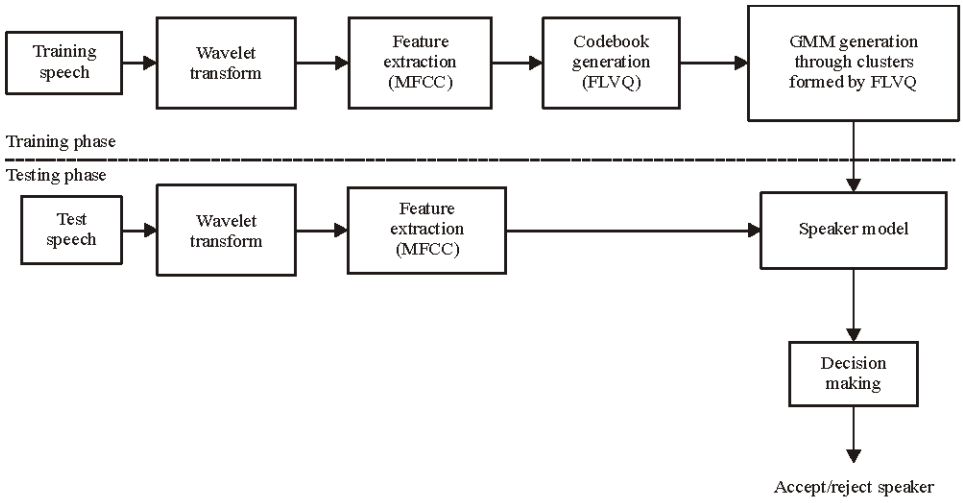

Fig. 1: Proposed speaker verification system using DWT-MFCC and FLVQ in GMM

Speaker modeling: In many speaker verification applications, accuracy and computational complexity are two major criteria for the selection of a proper system (Avci, 2007). State of the art system for speaker verification is Gaussian Mixture Model (GMM) based on the Maximum-Likelihood (ML) criterion which has been shown to outperform several other existing techniques. This is due to the fact that Gaussian mixture modeling is a powerful tool for representing virtually any distribution and the ability to form smooth approximations for many naturally occurring real world data. The multi modality is the most obvious property of GMM. The multi modality in data comes from multiple underlying causes each being responsible for one particular mixture component in the distribution. The model is described by mean, covariance and the mixture weights. The mean of the distribution describes the translation of the scatter from the origin. The covariance matrix describes the shape and orientation of the distribution of the data in space. In a GMM-based text-independent speaker verification system, generally a Universal Background Model (UBM) with a large number of Gaussian mixture components is created based on speech data from target and non-target speakers.

A Gaussian mixture density is a weighted sum of M component densities and is given by the Eq. 1:

$$f(\bar{x}/\lambda)=\sum_{i=1}^{M}p_i g_i(\bar{x}) \quad (1)$$

where, x is D-dimensional random vector, $g_i(\bar{x})$, i = 1,....M, are the component densities and p_i, i =1,....M are the mixture weights, λ is a model of a speaker. Each component density is further a D-variate Gaussian function:

$$g_i(\bar{x})=\frac{1}{(2\pi)^{D/2}|\Sigma_i|^{D/2}}\exp\left(\frac{1}{2}\left(\bar{x}-\bar{\mu_i}\right)^{T}\times\Sigma_i^{-1}\left(\bar{x}-\bar{\mu_i}\right)\right) \quad (2)$$

where, $\bar{\mu}_i$ is a D-dimensional mean vector, i =1,..., M; Σ_i is a DxD covariance matrix. The mixture weights satisfy the constraint that $\sum_{i=1}^{M}p_i=1$ which ensures the mixture is a true probability density function. Thus the set of parameters mean, covariance and mixture weights represents the model of a speaker $\lambda_i\{\bar{\mu}_i, \Sigma_i, p_i\}$.

GMM training using FLVQ algorithm: LVQ developed by Kohonen (1990) is used to determine the optimal codebook from the codebook initiated by any unsupervised clustering methods. LVQ is a supervised clustering technique which uses the class label for moving the code vectors in the optimum directions to improve the quality of classifier decision regions. In conventional LVQ training algorithm, weights are used to move the code vectors either towards the cluster center or away from the cluster center. The weights are updated during the learning algorithm. The weights are updated by:

$$w_j(new) = w_j(old)+\alpha[X-w_j(old)] \quad \text{if } T = V_j \quad (3)$$

$$w_j(new) = w_j(old)+\alpha[X-w_j(old)] \quad \text{if } T \neq V_j \quad (4)$$

If the test feature vector T, is closest to the jth cluster code vector V_j, the weight for the cluster 'j' is updated with the positive learning rate; otherwise the weight is updated with negative learning rate. This is crisp algorithm in which only the winning node is updated. The crisp change in weight vectors can be replaced by changes based on the fuzzy membership value which determines the distance between the winning prototype and the other vectors. The idea of fuzzy clustering is to divide the data into fuzzy partitions that overlap with one another. Therefore, the inclusion of data in a cluster is defined by a membership grade in [0, 1]. The learning algorithm for Fuzzy LVQ minimizes the loss function defined by Karayiannis *et al.* (1998).

$$L=\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{c}u_{ij}\left\|X_i-V_j\right\|^2 \quad (5)$$

where, $u_{ij} = u_j(X_i)$, $i\leq j\leq c$ are a set of membership values assigned to the prototypes V_j, $i\leq j\leq c$ with respect to each $X_i\in\chi$. The membership functions regulate the competition between the prototypes and the input by determining the strength of attraction between the input and the prototypes during the learning process. Assuming V_i is the winning prototype corresponding to the input vector X_i, i.e., V_i is the closest prototype to X_i in the Euclidean distance sense, the membership functions u_{ij}, $i\leq j\leq c$, can be of the form:

$$u_{ij}=\begin{cases}1 & \text{if } j=i\\ u\left(\dfrac{\left\|x-v_i\right\|^2}{\left\|x-v_j\right\|^2}\right) & \text{if } j\neq i\end{cases} \quad (6)$$

Using this membership function the loss function can be written as:

$$L_x=\sum_{j=1}^{c}u_{ij}\left\|x-v_i\right\|^2=\left\|x-v_i\right\|^2+\sum_{j\neq 1}^{c}u_{ij}\left\|x-v_j\right\|^2 \quad (7)$$

In such a case, the loss function measures the locally weighted error of each input vector with respect to the winning prototype. Minimization of Eq. 5 using gradient descent is difficult if the loss function Eq. 7 is defined with respect to the winning prototype because the winning prototype must be determined with respect to each input vector. It requires sequential updating of the prototypes with respect to the input vectors.

A variety of fuzzy algorithms for learning vector quantization can be derived by minimizing the loss function Eq. 7 using gradient descent. If X_i is the input vector, the winning prototype V_i, can be updated according to Karayiannis (1997) and Karayiannis *et al.* (1998) is:

$$\Delta \upsilon_i = \eta(x - v_i)\left(1 + \sum_{r \neq i}^{c} w_{ir}\right) \quad (8)$$

Where:

$$w_{ir} = w\left(\frac{\|x - v_i\|^2}{\|x - v_r\|^2}\right) \quad (9)$$

with w(z) = u'(z). Each non winning prototype $v_j \neq v_i$ can be updated by:

$$\Delta v_j = \eta(x - v_j)n_{ij} \quad (10)$$

Where:

$$n_{ij} = n\left(\|x - \upsilon_i\|^2 / \|x - \upsilon_j\|^2\right) \quad (11)$$

with n(z) = u(z)-zω(z). The update of the prototype depends on the learning rate $n \in [0, 1]$ which is a monotonically decreasing function of the number of iterations 'n'. The learning rate can be a linear function of n defined as:

$$\eta = \eta(n) = \eta_0\left(1 - \frac{n}{N}\right) \quad (12)$$

where, η_0 is its initial value and N the total number of iterations predetermined for the learning process. If X_i is the input vector, then the winning prototype is updated by Eq. 10 with w_{ir} evaluated in terms of the inference function as Eq. 9.

The non-winning prototypes $\upsilon_j \neq \upsilon_i$, can be updated by Eq. 18 with n_{ij} evaluated in terms of the inference function as Eq. 16.

The resulting algorithm can be summarized as follows:

- Initial codebook vectors are randomly generated from the data as $V_0 = [\upsilon_{1,0}, \upsilon_{2,0}, ..., \upsilon_{c,0}]$
- For each input vector x, the winning prototype is found such that:

$$\|x - v_{i,n-1}\|^2 < \|X - v_{j,n-1}\|^2, \forall j \neq i \quad (13)$$

where, n represents the current iteration

- The membership value of the non-winning prototypes are calculated as:

$$u_{ir,v} = u\|x - v_{i,n-1}\|^2 / \|x - v_{r,n-1}\|^2, \forall r \neq i \quad (14)$$

- The inference function used for the updating the winning prototype is calculated as:

$$w_{ir,v} = u'\|x - v_{i,n-1}\|^2 / \|x - v_{r,n-1}\|^2, \forall r \neq i \quad (15)$$

- The inference function used for the updating the non-winning prototype is calculated as:

$$n_{ir,v} = u_{ir,v} - \left(\frac{\|x - v_{i,n-1}\|^2}{\|x - v_{j,n-1}\|^2}\right) w_{ir,v} \forall r \neq i \quad (16)$$

- The winning prototype v_i is updated by:

$$v_{i,v} = u_{ir,v} + \eta(x - v_{i,n-1})\left(1 + \sum_{r \neq i}^{c} w_{ir,n}\right) \quad (17)$$

- The non-winning prototypes are updated by $v_j \neq v_i$ by:

$$v_{j,v} = u_{j,n-1} + \eta(x - v_{j,n-1}), n_{ij,v} \quad (18)$$

- The parameters of Gaussian mixture model (mean, covariance and weight) are calculated for each cluster formed by the codebook vectors
- The learning rate is updated as Eq. 12. If current iteration 'n' < N, then the process will be continued from step 2 for the updated codebook
- Finally, the Gaussian mixture model will be created for a speaker with number of mixtures equal to the size of codebook defined

Experimental setup: The proposed modeling method of GMM through FLVQ algorithm has been evaluated with the TIMIT database. The TIMIT database contains wideband (8 kHz) speech signals and is recorded in quiet environment. To reduce the total amount of required memory and experimental time, only a subset of the TIMIT database consisting of 49 (31 male and 18 female speakers from dialect region 1) speakers were used. Eight sentences (three "si" and five "sx" sentences) were used for training and the rest three "sa" sentences were used for testing. The speech signal is preprocessed and wavelet transformed MFCC features are extracted. The analysis window size was 30 ms with 10 ms overlapping. MFCC and its first and second derivative coefficients were calculated from each frame of the signals to compose a feature vector. From the feature vectors, a codebook (of size: 16, 32, 64, 128, 256) is created for every speaker using the proposed FLVQ algorithm. These codebook vectors which are the cluster centers are used to represent a Gaussian mixture model. The mean of a cluster is obtained by calculating the mean of all the vectors assigned to that particular cluster. The weights are determined by calculating the proportion of the vectors assigned to the cluster and the covariance matrix is the covariance matrix of the assigned vectors. Hence a Gaussian model with number of mixture components equal to the size of codebook is created for every speaker.

Initially, the state of art technique, GMM-UBM method of speaker modeling is developed and performance comparison is made with different number of mixtures in GMM.

In this study, two different experiments are conducted. First experiment is based on GMM trained using FLVQ algorithm. In the second experiment, codebook vectors obtained through FLVQ algorithm are directly used as reference vectors. In the first method, initially, a universal background model is developed for the feature vectors extracted from speech utterances of all 49 speakers. Then, GMM is developed for every target speaker through MAP adaptation from UBM-GMM. Verification is performed by applying the test feature vector to both UBM and the claimed target speaker model and the Log Likelihood Ratio (LLR) is calculated.

The log likelihood score is computed as:

$$\text{Log likelihood score} = \sum_{i=1}^{T} \log \left\{ \sum_{k=1}^{M} p_k g_k(x) \right\} \quad (19)$$

Then the ratio between these two scores is calculated to decide whether the claimed speaker is correct or not. If the Log Likelihood Ratio (LLR) value exceeds the predefined threshold then the claimed one will be accepted; otherwise he/she will be rejected.

In the second method, the Euclidean distance between the test feature vector and the codebook vectors are calculated. The testing is performed based on the closeness of the test vector with codebook vectors. Though the computation required for testing is less when compared to LLR calculation from GMM, the performance is poor.

Open set speaker verification is a more difficult problem than identification (Campbell, 1997). It involves finding whether the speech vectors are coming from the claimed speaker (whose model is known) or from an imposter (whose model is unknown). The result is a binary decision based on some score i.e., to accept the speaker or reject him/her as an imposter. Two kinds of errors can occur in this decision-making:

- **False Acceptance Rate (FAR):** Accepting an imposter as claimed speaker
- **False Rejection Rate (FRR):** Rejecting the true speaker as imposter

RESULTS AND DISCUSSION

Experiments were conducted to compare the performance of the speaker verification system trained with EM and FLVQ based GMM algorithm. The number of mixtures is fixed as 128 for the GMM based ASV. The system is trained with the help of the EM algorithm specified in (Barras and Gauvain, 2003; Reynolds *et al*., 2000). The training vectors are generated from TIMIT corpus. MFCC and DWT based MFCC features are computed from the speech samples. The results of the GMM based speaker verification system with the DWT-MFCC feature vectors trained with EM algorithm is reported in the Table 1.

Similarly, another GMM based ASV with the same number of mixtures is constructed and trained with the proposed FLVQ algorithm. The performance of the speaker verification system with the proposed method using FLVQ-GMM is compared with the above mentioned EM-GMM based ASV in terms of percentage of verification rate and equal error rate. From the comparison made in Table 1, it is proved that the performance of FLVQ based GMM is superior to the conventional GMM-EM method. The use of FLVQ in determining the parameters of GMM in ASR systems has been shown to be an improvement over EM algorithm for GMM techniques.

Table 1: Comparison between FLVQ-GMM and EM-GMM

Method of training GMM	VR (%)	FAR (%)	FRR (%)	EER
FLVQ algorithm	99.2	1.0	0.8	0.1
EM algorithm	96.8	5.2	3.2	0.5

Table 2: Comparison between various VQ codebook design and verification based on average distance

	Verification rate (%)				
Codebook based on	16	32	64	128	256
FLVQ	89.75	92.35	95.25	97.45	92.15
K-means	69.00	82.00	90.00	96.50	93.50
LVQ	83.00	90.00	93.00	98.15	94.50
Fuzzy C-means	88.53	85.01	86.01	89.95	85.42
LBG	65.20	77.25	85.12	87.35	82.11

In this study, for the second method of experiments, FLVQ clustering algorithm is used to develop the codebook. The working principle of FLVQ is different from K-means VQ and LBG VQ, in the sense that the soft decision making process is used while designing the codebooks in FLVQ, whereas in K-means VQ and LBG VQ the hard decision process is used. Moreover, in K-means VQ and LBG VQ each feature vector has an association with only one of the clusters. It may be difficult to come to a conclusion that the feature vector belongs to a particular cluster. Whereas, in FLVQ each feature vector has an association with all of the clusters with certain degrees of association, dictated by the membership function. In FLVQ, all of the feature vectors are associated with all of the clusters and therefore, there are relatively more feature vectors within each cluster and hence, the representative vectors i.e., the code vectors may be more reliable than for the other VQ techniques. Therefore, clustering may found to be improved when using FLVQ. The use of FLVQ in determining the codebook in ASR systems has been shown to be an improvement over other VQ techniques.

From Table 2, the performance of the proposed SR system is shown to be an improvement over the other systems studied. Even though the performance of fuzzy C-means is comparable to FLVQ for small size codebooks, FLVQ performs better for large size codebooks also. The proposed feature extraction algorithm outperformed the feature extraction methods reported in the literature.

CONCLUSION

Training of gaussian mixture model using fuzzy learning vector quantization provides high quality cluster centers and in turn efficient determination of parameters of GMM. Compared with conventional algorithm (EM) for GMM, the proposed algorithm performs better even for large number of speakers. Though the performance of FLVQ is comparatively high, the computation time is more than that for GMM. This is because of the requirement of computation of membership function parameters at each iteration and also the time required for convergence.

REFERENCES

Avci, E., 2007. An automatic system for Turkish word recognition using discrete wavelet neural network based on adaptive entropy. Arabian J. Sci. Eng., 32: 239-250.

Barras, C. and J. Gauvain, 2003. Feature and score normalization for speaker verification of cellular data. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, April 6-10, 2003, Volume 2, Hong Kong, pp: 49-52.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, Norwell, MA., USA., Pages: 256.

Bezdek, J.C., R. Ehrlich and W. Full, 1984. FCM: The fuzzy c-means clustering algorithm. Comput. Geosci., 10: 191-203.

Campbell, J.P. Jr., 1997. Speaker recognition: A tutorial. Proc. IEEE., 85: 1437-1462.

Equitz, W.H., 1989. A new vector quantization clustering algorithm. IEEE Trans. Acoust. Speech Signal Process., 37: 1568-1575.

Gray, R.M., 1984. Vector quantization. IEEE ASSP Mag., 1: 4-29.

He, J., L. Liu and G. Palm, 1997. A new codebook training algorithm for VQ-based speaker recognition. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 2, April 21-24, 1997, Munich, pp: 1091-1094.

Ho, D.W., P.A. Zhang and J. Xu, 2001. Fuzzy wavelet networks for function learning. IEEE Trans. Fuzzy Syst., 9: 200-211.

Karayiannis, N.B. and P.I. Pai, 1996. Fuzzy algorithms for learning vector quantization. IEEE Trans. Neural Networks, 7: 1196-1211.

Karayiannis, N.B., 1997. A methodology for constructing fuzzy algorithms for learning vector quantization. IEEE Trans. Neural Networks, 8: 505-518.

Karayiannis, N.B. and J.C. Bezdek, 1997. An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. IEEE Trans. Fuzzy Syst., 5: 622-628.

Karayiannis, N.B., P.I. Pai and H. Zervos, 1998. Image compression based on fuzzy algorithms for learning vector quantization and wavelet image decomposition. IEEE Trans. Image Process., 7: 1223-1230.

Kohonen, T., 1990. The self-organizing maps. Proc. IEEE, 78: 1464-1480.

Kummamuru, K., A. Dhawale and R. Krishnapuram, 2003. Fuzzy co-clustering of documents and keywords. Proceedings of the 12th IEEE International Conference on Fuzzy Systems, Volume 2, May 25-28, 2003, St. Louis, USA., pp: 772-777.

Linde, Y., A. Buzo and R.M. Gray, 1980. An algorithm for vector quantizer design. IEEE Trans. Commun., 28: 84-95.

Reynolds, D.A., T.F. Quatieri and R.B. Dunn, 2000. Speaker verification using adapted gaussian mixture models. Digital Signal Process., 10: 19-41.

Tsao, E.C.K., J.C. Bezdek and N.R. Pal, 1994. Fuzzy Kohonen clustering networks. Pattern Recognit., 27: 757-764.

Zhang, Q. and A. Benveniste, 1992. Wavelet networks. IEEE Trans. Neural Networks, 3: 889-898.