International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 6, June-2015, pp: (578-587), Impact Factor: 1.252, Available online at: www.erpublications.com

Speaker Recognition using Fuzzy Integral

Prateek Sangwan[1], Shamsher Malik[2]
[1]UIET MDU, Rohtak, Haryana, India
[2]Asstt. Prof., UIET MDU, Rohtak, Haryana, India

Abstract: **Speaker Recognition is a process of automatically recognizing who is speaking on the basis of the individual information included in speech waves. Speaker Recognition is one of the most useful biometric recognition techniques in this world where insecurity is a major threat. There are a lot of feature matching techniques used in speaker recognition such as Vector Quantization (VQ), Gaussian Mixture Model (GMM), Support Vector Machine(SVM), Hidden Markov Modeling (HMM). The major technique for speaker recognition is based on MFCC (Mel-Frequency Cepstral Coefficients) and GMM. But even the technique based on MFCC and GMM are known to perform very well only for small population speaker recognition under low-noise conditions. So, we present one more technique for feature matching which may able to overcome above limitations. In this paper, we explain fuzzy integral based feature matching technique which we have implemented in MATLAB on small level. In this technique, we combine the concepts of choquet fuzzy integral and back propagation algorithm to implement feature matching module. In this paper, we also provide architectural description of modules which helps in understanding fuzzy integral based feature matching technique practically. Those architecture diagrams also helps in understanding how those modules are interlinked. Finally, we present the experimental results obtained by executing coding of proposed technique on small databases.**

Keywords: Mel-Frequency cepstral coefficients (MFCC), Gaussian mixture model (GMM), Fuzzy integral, Feature vector, Back propagation algorithm, Choquet fuzzy integral, Information source.

I. INTRODUCTION

Speaker recognition is an important branch of speech processing. It is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves as shown in figure 1 [4].

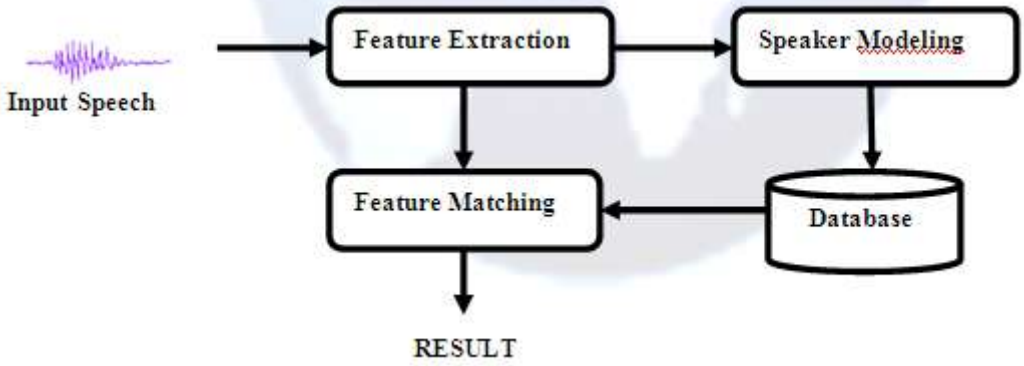

Figure 1: Speaker Recognition process

The major technique for speaker recognition is based on MFCC (Mel-Frequency Cepstral Coefficients) and GMM (Gaussian Mixture Model). Another emerging technique which becomes very population is the i-vector approach (including the joint factor analysis approach). The i-vector approach usually requires a large number of data to perform well and the computational complexity can be high when applying i-vector to speaker identification especially for large population case [1]. In our paper, we evaluate the performance of fuzzy integral along with learning algorithm as a feature matching technique which we have implemented in MATLAB by using four MFCCs for each speaker and also with some variations in those MFCCs. The approaches based on MFCC and GMM are known to perform very well for small population speaker recognition under low-noise conditions. However, they also have some drawbacks. The first drawback is that they suffer from the mismatch between training and testing caused by noisy conditions. The noisy conditions can severely degrade the identification performance. The second drawback is actually a common problem of almost all existing speaker recognition techniques [1].

The success of almost all existing identification systems (including GMM-based systems) lies in the fact that they are

trained on datasets with only a relatively small population. However, it is pretty straightforward that when the population has a significant increase (e.g., thousands of registered speakers or even more), the probability of identification errors will also increase. The experiments mainly conduct when training and testing conditions are matched without additive noise or channel variations. Therefore, the population becomes an extremely important impact factor of the identification performance under noisy conditions [1]. In this paper, we present a fuzzy integral based feature matching technique which we have implemented in MATLAB on small level and try to provide another feature matching technique. The key idea of our approach is that we use fuzzy integral for decision-making and learning algorithm to develop capability in system to learn "how to recognize speaker" by using training phase database.

2. Background and Related Information

Before we proceed towards the fuzzy integral and training algorithms which we used in the speaker recognition system in our work, a brief background and related information on various topics like fuzzy logic , membership function, neural network etc. is presented in this section.

2.1 Various processes involve in ASR system

The various processes involve in the working of automatic speaker recognition system are the following:

2.1.1 Preprocessing

The speech signal needs to undergo various signal conditioning steps before being subjected to the feature extraction methods. Pre-processing the signal reduces the computational complexity while operating on the speech signal. These tasks include:-

- Truncation
- Frame blocking
- Windowing
- Fast Fourier Transform

2.1.2 Feature extraction

Feature extraction is the process that extracts a small amount of data from the speaker's voice signal that can later be used to represent that speaker. Many feature extraction techniques are available, some of them are:-

- Mel-frequency cepstral coefficients (MFCC)
- LPC-based cepstral parameters
- Relative spectra filtering of log domain coefficients (RASTA)
- Local discriminant bases (LDB)

2.1.3 Feature matching

Feature matching is a classification procedure to classify objects of interest into one of a number of classes. The objects of interest are called patterns which are sequences of feature vectors that are extracted from an input speech using the MFCC processor. Each class here refers to each individual speaker. There are a lot of feature matching techniques used in speaker recognition such as:

- Vector Quantization (VQ)
- Gaussian Mixture Model (GMM)
- Support Vector Machine (SVM)
- Hidden Markov Modeling (HMM)

In our work, we try to add one more technique to this list and that is fuzzy integral based feature matching technique.

2.2 Membership function

A fuzzy set is defined by a function that maps objects in that fuzzy set to their membership value. Such a function is called the membership function. It is denoted by 'μ'. The membership function of a fuzzy set 'A' is denoted by 'μ_A'. The membership value of 'x' in 'A' is denoted as '$\mu_A(x)$' [5]. The value of membership function is [0,1]. There are three cases possible according to the value of $\mu_A(x)$:

1) If $\mu_A(x) = 0$
   then, 'x' will entirely not present in fuzzy set A.
2) If $\mu_A(x) = 1$

then, 'x' will be completely present in fuzzy set A.

3) If $0 < \mu_A(x) < 1$
then, 'x' will be partially present in fuzzy set A.

2.3 Neural network

Neural networks store information in the strengths of the interconnections. In a neural network new information is added by adjusting the interconnection strengths, without destroying the old information. As information is stored in the connections and it is distributed throughout, the network can function as a memory. This memory is content addressable, in the sense that the information may be recalled by providing partial or even erroneous input pattern. The information is stored by association with other stored data like in the brain. Because of the inherent redundancy in information storage, the networks can also recover the complete information from partial or noisy input pattern [6].

2.3.1 Learning

The weights are adjusted to learn the pattern information in the input samples. Typically, learning is a slow process, and the samples containing a pattern may have to be presented to the network several times before the pattern information is captured by the weights of the network. A large number of samples are normally needed for the network to learn the pattern implicit in the samples. Pattern information is distributed across all the weights, and it is difficult to relate the weights directly to the training samples.. Another interesting feature of learning is that the pattern information is slowly acquired by the network from the training samples, and the training samples themselves are never stored in the network. That is why we say that we learn from examples, not store the examples themselves [6].Learning laws describe the weight vector for the ith processing unit at time instant (t + 1) in terms of the weight vector at time instant (t) as follows:

$$w_i(t + 1) = w_i(t) + \Delta w_i(t) \qquad (2.1)$$

where, $\Delta w_i(t)$ is the change in the weight vector.

2.4 Decision making using fuzzy integral

Number of decision making systems have been designed for the aggregation of multiple sets of supporting or conflicting evidence. There are number of methods for combining the information from multiple sources. Some of them are Bayesian reasoning, Dempster-Sharer theory, and fuzzy set techniques. A recent addition to above list is fuzzy integral. The fuzzy integral differs from the other methods in that it considers both the evidence supplied by each information source and the expected worth of each subset of sources in its decision making process. It is a non-linear combination of the objective evidence which is present in the form of a fuzzy membership function, along with the worth of subsets of sources w.r.t. the decision.

3. Fuzzy Integral Based Feature Matching Technique

The fuzzy integral combines information from multiple sources in order to achieve a final classification. A fuzzy integral is calculated for every classification hypothesis, and then the integral having the largest value usually indicates the class label [2]. Let $O=\{O_1, O_2, O_3, \ldots, O_M\}$ be a set of instances of objects from some data media (video, infrared video, range data, etc.). Each instance 'O_k' may be a single occurrence of an object in the set O, or it may only be a single instance of multiple occurrences of the same object, say through time. We will refer to an object instance simply as an object.

Let these objects be separable into classes $C_1, C_2 \ldots\ldots C_n$, where each class not only contains all instances of a particular object but also contains other objects which we define to be in the same class. Given an object, each class C_j will represent a hypothesis that the information obtained from the object was generated by an object in that class. The set of class hypotheses is the decision which is to be resolved (i.e., the class label of this object). Let $X = \{x_1, x_2 \ldots\ldots, x_m\}$ be a finite set which represents a set of 'm' information sources. Each source 'x_i' may be a feature (statistic, texture, shape, or other) which can be calculated from an object instance 'O_k'. An information source can also be the output of an algorithm which fuses the information from a group of sources, or any other type of information source, say context or intelligence data [2].

Given an object 'O_k' and a class hypothesis 'C_j', let $h_j^k : X \rightarrow [0, 1]$ be a function from X to the closed interval [0, 1]. The h-function is defined for each information source 'x_i' in X. In simple words, we can say that $\mathbf{h_j^k(x_i)}$ represents that to which extend **'O_k'** belongs to class **'C_j'** from the standpoint of information source **'x_i'**.

Consider set of information sources 'X', and let $g^i = g(\{x_i\})$. The mapping $x_i \rightarrow g^i$ will be called a fuzzy density function. The fuzzy density value gi is interpreted as the importance of the information source x_i in determining the evaluation of a class hypothesis. A set of fuzzy density values can be constructed for the information sources in the set X. The value of 'λ' for any Sugeno fuzzy measure can be uniquely determined [2] for a finite set X using Eqn(4.1) and the facts that $X = \bigcup_{i=1}^{n}\{x_i\}$ and g(X) = 1, which leads to solving the following equation for λ:

$$1 + \lambda = \prod_{i=1}^{n}(1 + \lambda . g_i) \quad (3.1)$$

This is a polynomial in 'λ' of degree ' n-1' and root of this polynomial will give the required value of 'λ'.

There are a number of interesting families of fuzzy integral but the fuzzy integral which we consider in our work is the Choquet fuzzy integral.

3.1 Choquet fuzzy integral

Assume $h(x_1)$, $h(x_2),\cdots$, $h(x_m)$ are the evidence provided by the input sources x_1, $x_2,\cdots,x_m$ and g is a λ fuzzy measure, then we can denote Choquet fuzzy integral as [3] :-

$$f = \int_x h(\cdot) \text{ o } g(\cdot)$$

For a finite set of X , the Choquet fuzzy integral can be computed as follows:

$$f = \sum_{i=1}^{m} h(x_i)[g(A_i) - g(A_{i+1})] \quad (3.2)$$

3.1.1 Choquet fuzzy integral based network

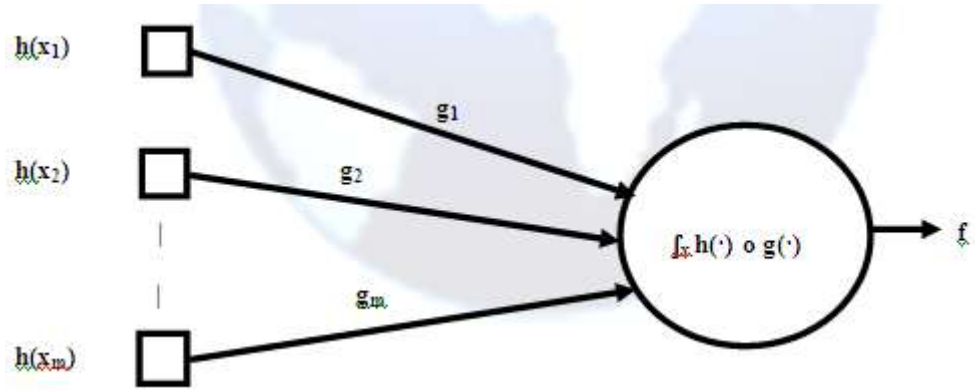

Figure 2: Architectural graph of a fuzzy integral-based neural node

The fuzzy integral-based network is a directed graph consisting of neural nodes with interconnecting synaptic links and which is characterized by following four properties [3]:-

- Each neuron is represented by a set of linear synaptic links and a fuzzy integral function with respect to certain fuzzy measure.
- The synaptic links of a neuron (called fuzzy densities) is interpreted as the degree of importance of the respective input signals.
- The weighted computation of the input signals defines the activity level of the neuron, which is the output value.
- The output level is restricted to the range between the minimum and maximum level of the input signals via the fuzzy integral function.

3.2 Back propagation learning algorithm

In our work, we are applying back propagation learning algorithm in our single layer fuzzy integral-based neural network which is nothing but internal structure of feature matching module. The back propagation learning involves propagation of the error backwards from the output layer to the hidden layers in order to determine the update for the weights leading to the units in a hidden layer. The error at the output layer itself is computed using the difference between the desired output and the actual output at each of the output units. There is no feedback of the signal itself at any stage, as it is a feed forward neural network.

Let us assume that there are 'm' inputs to an output node and the training data for this node consists of sets of 'M' inputs $h^k(x_1)$, $h^k(x_2),\cdots$, $h^k(x_m)$ with 'M' corresponding desired outputs y^k (for k=1,2,3,$\cdots$,M). The back propagation algorithm is used for the learning process. The learning process is to determine the best set of fuzzy densities for this node in such a way that the discrepancy between the desired and actual fuzzy integral behavior is minimized. One measure that is commonly used as discrepancy is the sum of squared error by:

$$E = \sum_K E_K = \sum_K [\sum_j (f_j{}^k - y_j{}^k)^2] \quad (3.3)$$

'n' is the number of the output nodes, 'm' is the number of the input nodes and g_{ji} is the synaptic weight connecting i^{th} input node to j^{th} output node. The network is then optimized by minimizing 'E' with respect to the synaptic weights (fuzzy densities) of the network. Thus, we update the densities using the following equations based on gradient descent:

$$g_{ji}^{new} = g_{ji}^{old} + \Delta g_{ji} \quad (3.4)$$

Figure 3: Representation of fuzzy density values

For training a feedforward neural network, we use the following estimate of the gradient descent along the error surface to determine the increment in the weight connecting i^{th} input node and j^{th} output node :

$$\Delta g_{ji} = -\eta \frac{\partial E}{\partial g_{ji}} \quad (3.5)$$

On differentiating equation(3.8) and put into above equation, we get :

$$\Delta g_{ji} = -2\,\eta\,[\,\sum_{k=1}^{M}(f_j{}^k - y_j{}^k)\frac{\partial f_j{}^k}{\partial g_{ji}}\,] \quad (3.6)$$

During training the neural network, the training patterns are applied in some random order one by one, and the weights are adjusted using the back propagation learning law. Each application of the training set patterns is called a cycle. The patterns may have to be applied for several training cycles to obtain the output error to an acceptable low value. The main objective is to capture the implicit pattern behaviour in the training set data so that adequate generalization takes place in the network. The generalization feature is verified by testing the performance of the network for several new (test) patterns [6].

4. Architectural Description of Modules

Speaker recognition system can be divided into three main modules:-

1. Feature extraction module
2. h -function module
3. Feature matching module

In our work, we have implement only h-function module and feature matching module.

4.1 Architecture of feature extraction module

An object is applied to 'm' number of feature extraction blocks as an input. Each block provides one mel frequency cepstrum coefficient (mfcc) of a given object. That is, output of feature extraction module consist of 'm' number of coefficients.

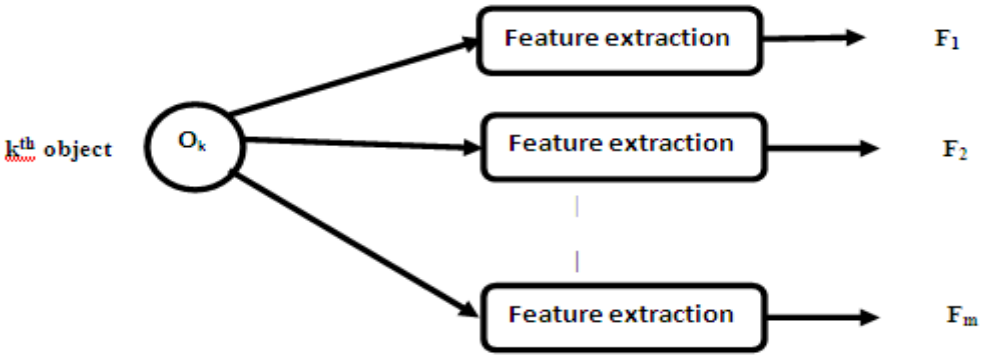

Figure 4: Architecture for feature extraction module

4.2 Architecture of h-function module

The output of feature extraction module ,i.e, {F_1, F_2, F_3,⋯⋯, F_m} is applied to h-function module as input.Each class has its own h-function module. This block behaves like an interfacing device between feature extraction and feature matching modules. The value of h-function module outputs has some practical meaning which we will see later.

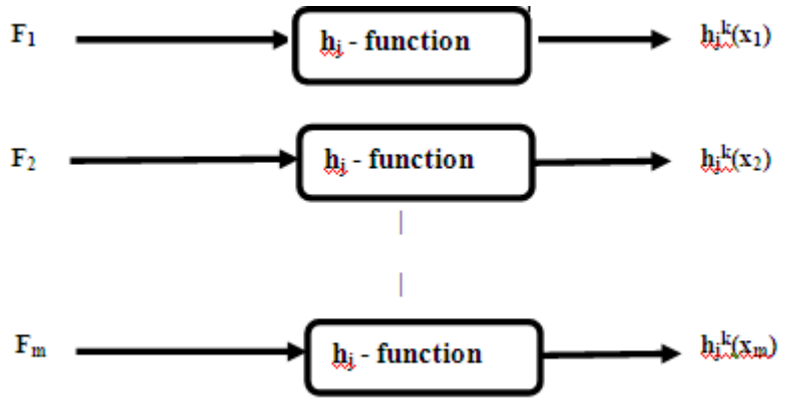

Figure 5: Architecture of h-function module for j^{th} class

4.3 Architecture of Feature matching module

The functioning of feature matching module in our work depends on choquet fuzzy integral (take place inside of each output node) for decision making process and back propagation learning algorithm as a training algorithm. The inputs for this module are nothing but outputs of h-function modules. In this, there are 'n' number of outputs nodes, where, 'n' is no of classes. Each output node corresponds to a particular class. In the last step of the feature matching module, that output node which gives maximum value as a output among all the output nodes will indicate the class of a given object.

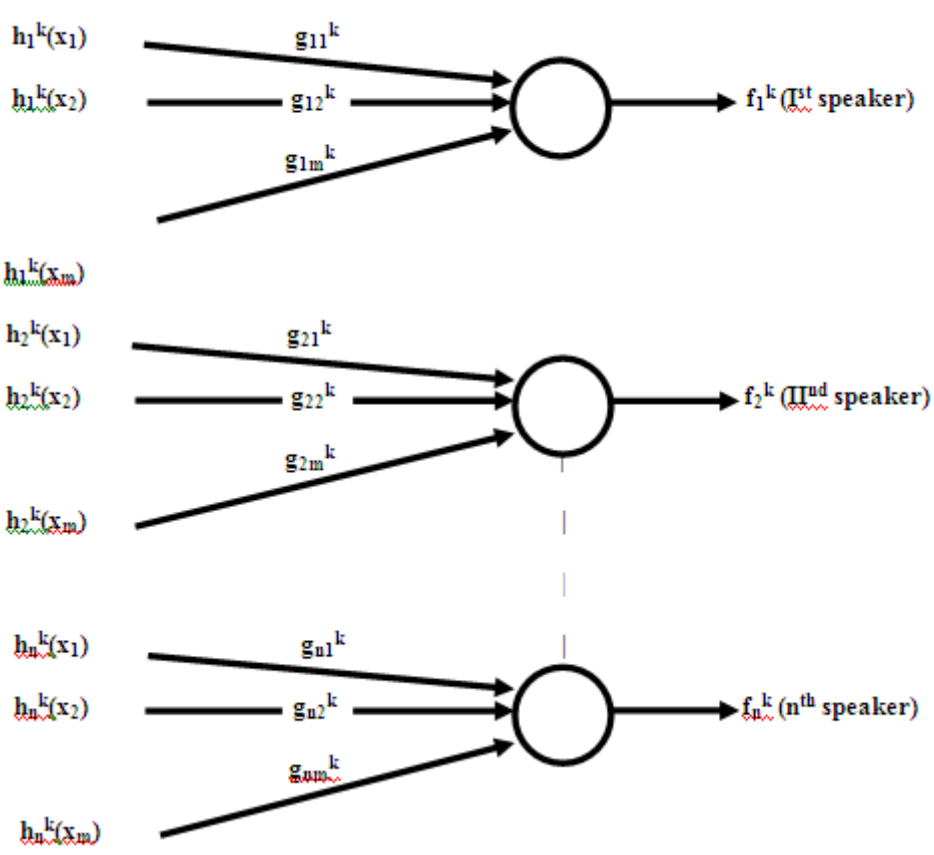

Figure 6: Architecture of feature matching module

4.4 Some important points

- Classes in above discussion are registered speakers whose mel frequency cepstrum coefficients are already present in the database.
- Objects in training pattern set are those speakers whose mel frequency cepstrum coefficients are used to train the neural network.In other words, those objects whose class is already known.

- Objects in testing pattern set are those speakers which have to be recognized from all the speakers whose coefficients are present in database.

5. EXPERIMENT RESULTS

In this chapter, the speaker recognition system which was discussed in the previous chapters will be analysed for its h-module and feature matching module by taking four MFCCs of 11 speakers as a content of our database as shown below.

5.1 Database of speakers

The following table contains the MFCCs of 11 speakers which we use to analyse the performance of our coding for h-function and feature matching module.

Table 5.1 Database contains the MFCCs of 11 speakers

	Coefficient 1	Coefficient 2	Coefficient 3	Coefficient 4
Speaker 1	-27.7020	0.2300	-0.6928	0.0862
Speaker 2	-28.0880	-0.0324	-1.0803	-0.3086
Speaker 3	-39.1189	-5.4464	0.5683	-0.7025
Speaker 4	-41.4200	-7.7150	0.8154	0.1369
Speaker 5	-43.8300	-2.3321	-0.4223	0.0652
Speaker 6	-45.6500	-1.7834	0.4581	-0.4312
Speaker 7	-48.2300	0.7823	0.6514	-0.6518
Speaker 8	-50.5800	-4.2414	0.5547	-0.5421
Speaker 9	-52.8700	-9.8314	-0.9823	0.1162
Speaker 10	-59.7800	-6.2813	-0.4512	0.0952
Speaker 11	-55.2300	-3.2817	0.3818	0.0782

5.2 Database of speakers with some variation

The following table contains the MFCCs of speakers with some variation in their values to analyse the system performance. Because it is possible that MFCCs of speakers are not same exactly whenever they speak.

Table 5.2 MFCCs of speakers with some variation in their values

	Coefficient 1	Coefficient 2	Coefficient 3	Coefficient 4
Speaker 1	-26.7020	0.1900	-0.7528	0.0562
Speaker 2	-26.0880	-0.0124	-1.1203	-0.2786
Speaker 3	-40.1189	-4.5464	0.4683	-0.9025
Speaker 4	-41.9200	-7.9150	0.6954	0.1669
Speaker 5	-43.3300	-2.2321	-0.4923	0.0752
Speaker 6	-44.9500	-1.1034	0.4881	-0.4712
Speaker 7	-47.6300	0.8323	0.6114	-0.6918
Speaker 8	-49.6800	-4.1614	0.5047	-0.4721
Speaker 9	-53.5700	-9.2314	-0.8923	0.1962
Speaker 10	-58.9800	-6.2113	-0.4912	0.0892
Speaker 11	-56.0300	-3.2017	0.3118	0.0892

5.3 Training of feature matching module

The training allows the development of a valid set of fuzzy density values ' g_{ji}' even when no prior knowledge about the information sources is available. Intially, we take values of all fuzzy density values or weights equal to 0.2000 which indicate that, at this moment, system does not how to identify the speakers. So, for the training, we apply MFCCs of first eight speakers given in table 6.3 with outputs ,i.e, with the information of their speakers.

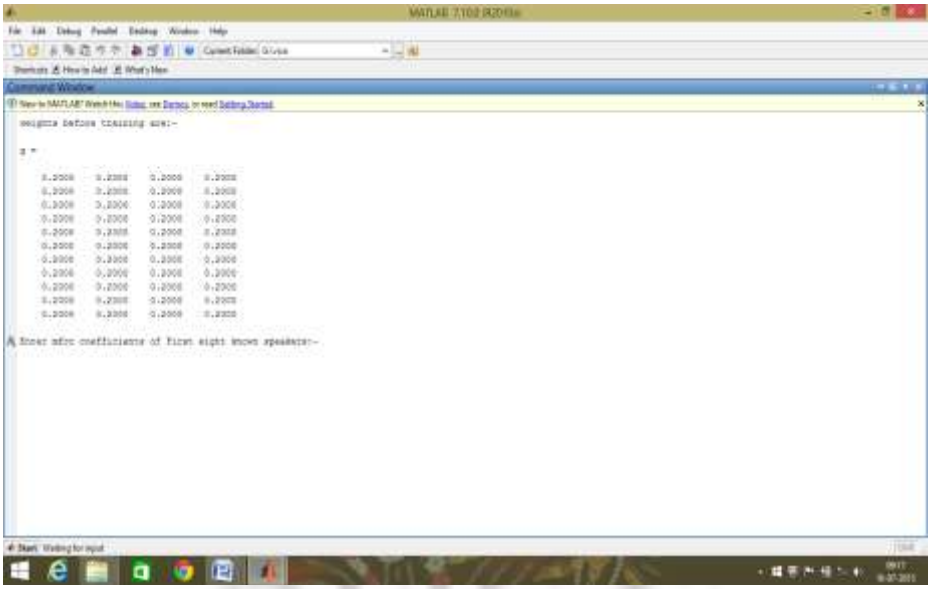

Figure 7: Fuzzy density values before training

Table 5.3 MFCCs of first eight speakers used for training

	Coefficient 1	Coefficient 2	Coefficient 3	Coefficient 4
Speaker 1	-26.7020	0.1900	-0.7528	0.0562
Speaker 2	-26.0880	-0.0124	-1.1203	-0.2786
Speaker 3	-40.1189	-4.5464	0.4683	-0.9025
Speaker 4	-41.9200	-7.9150	0.6954	0.1669
Speaker 5	-43.3300	-2.2321	-0.4923	0.0752
Speaker 6	-44.9500	-1.1034	0.4881	-0.4712
Speaker 7	-47.6300	0.8323	0.6114	-0.6918
Speaker 8	-49.6800	-4.1614	0.5047	-0.4721

After training, we get a optimal set of fuzzy density values or weights. Now, our system is already to identify speaker among those 11 speakers even when MFCCs of some speaker is not still applied to system during the training. So, fuzzy density values after training are :-

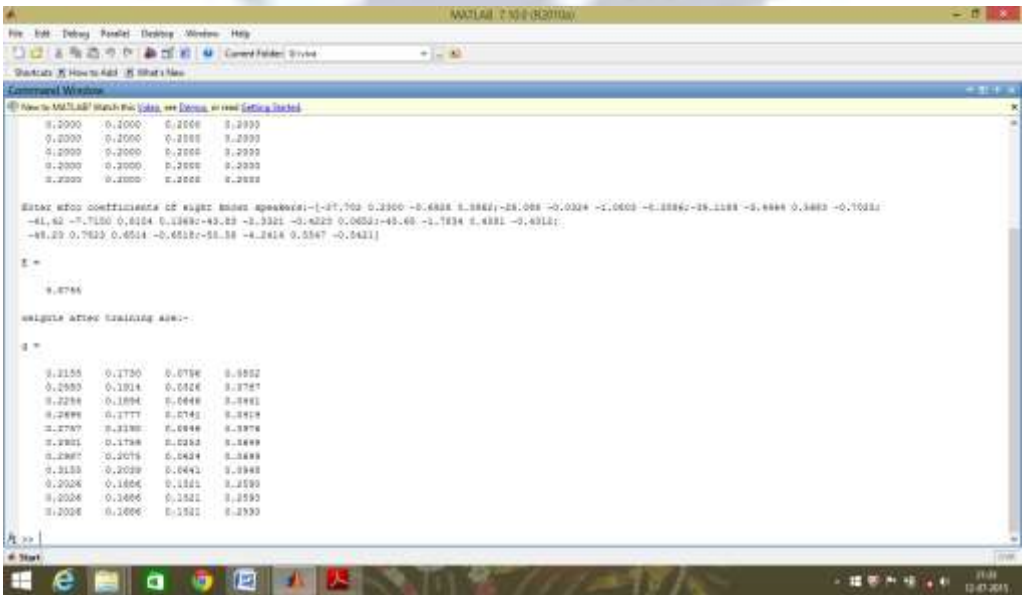

Figure 8: Fuzzy density values after training

$g_{ji} =$

0.2155	0.1730	0.0796	0.0802
0.2580	0.1914	0.0326	0.0787
0.2254	0.1894	0.0646	0.0441
0.2694	0.1777	0.0741	0.0419
0.2757	0.2190	0.0846	0.0976
0.2901	0.1759	0.0253	0.0649
0.2987	0.2075	0.0624	0.0699
0.3155	0.2039	0.0641	0.0948
0.2026	0.1686	0.1321	0.2590
0.2026	0.1686	0.1321	0.2590
0.2026	0.1686	0.1321	0.2590

Figure 9: Fuzzy density values after training

5.4 Testing of feature matching Module

In testing phase, we apply some exact MFCCs and approximate MFCCs of speakers as shown below in table 6.4 and observe its accuracy in the identifying the speakers.

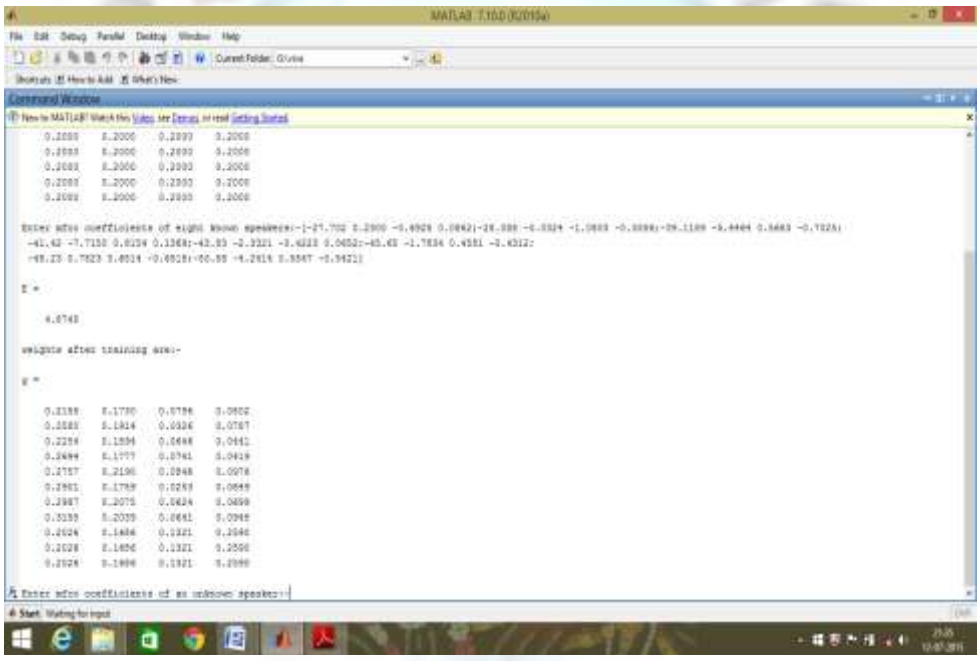

Figure 10: Command window during testing phase

Table 5.4 MFCCs used in testing phase

	Coefficient 1	Coefficient 2	Coefficient 3	Coefficient 4
Unknown Speaker 1	-26.7020	0.1900	-0.7528	0.0562
Unknown speaker 2	-48.2300	0.7823	0.6514	-0.6518
Unknown speaker 3	-50.5800	-4.2414	0.5547	-0.5421
Unknown speaker 4	-56.0300	-3.2017	0.3118	0.0892
Unknown speaker 5	-43.3300	-2.2321	-0.4923	0.0752
Unknown speaker 6	-44.9500	-1.1034	0.4881	-0.4712
Unknown speaker 7	-39.1189	-5.4464	0.5683	-0.7025
Unknown speaker 8	-52.8700	-9.8314	-0.9823	0.1162

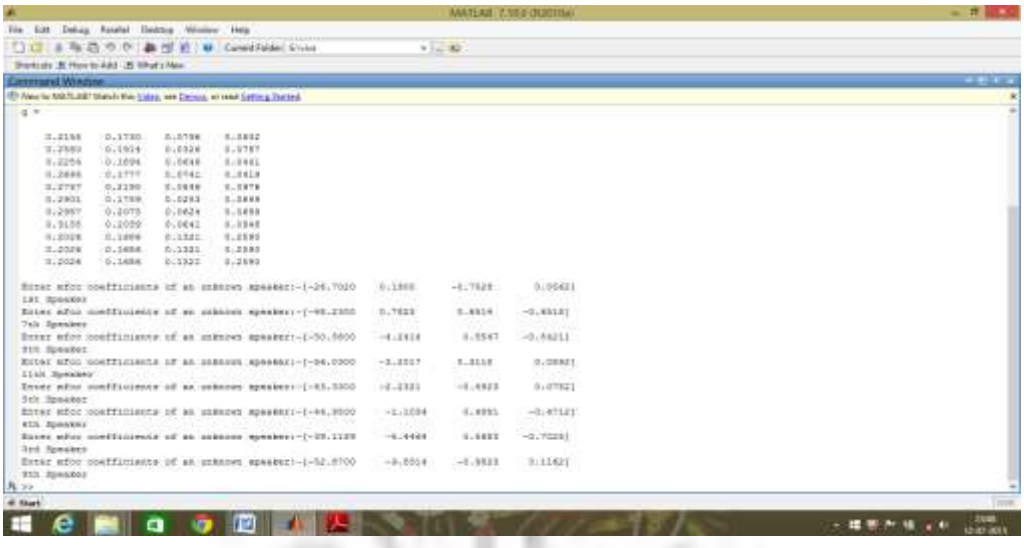

Figure 11: Command window showing results of testing phase

5.5 Testing phase results

Table 5.5 This table contains the results of testing phase

Inputs	Desired outputs	Actual outputs
Unknown Speaker 1	Speaker 1	Speaker 1
Unknown speaker 2	Speaker 7	Speaker 7
Unknown speaker 3	Speaker 8	Speaker 8
Unknown speaker 4	Speaker 11	Speaker 11
Unknown speaker 5	Speaker 5	Speaker 5
Unknown speaker 6	Speaker 6	Speaker 6
Unknown speaker 7	Speaker 3	Speaker 3
Unknown speaker 8	Speaker 9	Speaker 9

CONCLUSION

The presented work is the implementation of feature matching using choquet fuzzy integral and back propagation training algorithm in MATLAB. The results obtained during testing phase shows the Speaker recognition system can produce better results not only with exact MFCCs of speakers but also give good results with approximated MFCCs of speakers, when its feature matching module is implemented using fuzzy integral. So, the proposed method is truthful in recognizing the speakers through their MFCCs. From all these, we can conclude that choquet fuzzy integral and back propagation algorithm as a combination provides a good technique for feature matching.

REFERENCES

[1]. Yakun Hu, Dapeng Wu, Fellow, IEEE, and Antonio Nucci "Fuzzy-Clustering-Based Decision Tree Approach for Large Population Speaker Identification" IEEE Transaction on Audio, Speech, and Language processing, VOL. 21, NO. 4, APRIL, 2013.

[2]. James M. Keller and Jeffrey Osborn "Training the Fuzzy Integral" International Journal of Approximate Reasoning 1996; 15:1-24 © 1996 Elsevier Science Inc. .

[3]. Jung-Hsien Chiang, Member, IEEE "Choquet Fuzzy Integral-Based Hierarchical Networks for Decision Analysis" IEEE Transactions on Fuzzy Systems, VOL. 7, NO. 1, February, 1999.

[4]. Izuan Hafez Ninggal & Abdul Manan Ahmad "The Fundamental of Feature Extraction in Speaker Recognition : A Review" Proceedings of the Postgraduate Annual Research Seminar 2006.

[5]. John Yen, Reza Langari "Fuzzy Logic intelligence,control and information" Pearson Education,1999.

[6]. B.Yegnanarayana "Artificial Neural Networks" PHI Learning Private Limited, 2010.