

# How to Reduce Dimension while Improving Performance

Abdelghani Harrag<sup>1\*</sup>, D. Saigaa<sup>1</sup>, A. Bouchelaghem<sup>1</sup>, M. Drif<sup>1</sup>, S. Zeghlache<sup>1</sup>,  
and N. Harrag<sup>2</sup>

<sup>1</sup>Department of Electronics, Faculty of Technology  
University Mohamed Boudiaf Msila  
BP 166 Ichbilia 28000 Algeria  
abdelghani.harrag@gmail.com

<sup>2</sup>Department of Informatics, Faculty of Sciences, University Ferhat Abbas Setif,  
El-Baz 19000 Setif Algeria

**Abstract.** This paper addresses the feature subset selection for an automatic Arabic speaker recognition system. An effective algorithm based on genetic algorithm is proposed for discovering the best feature combinations using feature reduction and recognition error rate as performance measure. Experimentation is carried out using QSDAS corpora. The results of experiments indicate that, with the optimized feature subset, the performance of the system is improved. Moreover, the speed of recognition is significantly increased, number of features is reduced over 60% which consequently decrease the complexity of our ASR system

**Keywords:** genetic algorithm; feature selection; speaker recognition.

## 1 Introduction

The speech signal is rich in information and redundancy. The redundancy is robust against background noise, distortion and damage suffered by the voice signal. This richness expresses the informations that are simultaneously conveyed by the message linguistic context, the anatomical features, the state and the socio-cultural constraints of the speaker.

Speech signals contain a huge amount of information and can be described as having a number of different levels of information. At the top level, we have lexical and syntactic features, below that are prosodic features, further below these are phonetic features, and at the most basic level we have low-level acoustic features, which generally give information on the system that creates the sound, such as the speakers' vocal tract. Information solely about how the sound is produced (from low-level acoustic features) should give enough information to identify accurately a speaker, as this is naturally speaker dependent and independent of text [1].

Low-level acoustic features also contain some redundant features, which can be eliminated using Feature Selection (FS) techniques. The objective of feature selection

---

\* Corresponding author.

is to simplify a dataset by reducing its dimensionality and identifying relevant underlying features without sacrificing predictive accuracy. By doing that, it also reduces redundancy in the information provided by the selected features [2]. In real world problems, feature selection is a must due to the abundance of noisy, irrelevant or misleading features. Selected features should have high inter-class variance and low intra-class variability. Ideally, they should also be as independent of each other as possible in order to minimize redundancy.

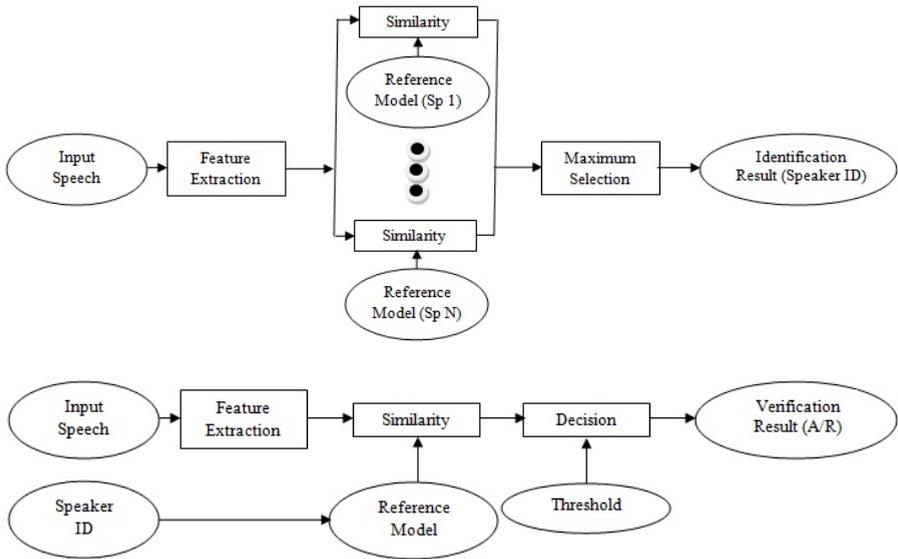
Feature selection is extensive and it spreads throughout many fields, including signal processing [3], face recognition [4], text categorization [5], data mining and pattern recognition [6]. Among many methods that are proposed for feature selection, population based optimization techniques [7][8] such as genetic algorithm have attracted a lot of attention. These methods attempt to achieve better solutions by application of knowledge from previous iterations. Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space prompting to use them. Because of their advantages, recently, GAs have been used as a tool for feature selection in data mining [9].

In this paper, we propose a GA-based algorithm for feature selection in VQ-based Arabic Speaker Recognition (ASR) system. We apply it to feature vectors containing Mel-Frequency Cepstral Coefficients (MFCCs), their first and second derivative. Then, feature vectors are applied to a VQ model followed by K-Nearest-Neighbor (KNN) classifier used to measure the performance of selected feature vector based on recognition rate and selected feature vector size. The rest of this paper is organized as follows. Section 2 presents the taxonomy of ASR systems. Genetic algorithms are described in Section 3. Section 4 reports discussion of the results obtained. The conclusion and future works are offered in the last section.

## 2 Automatic Speaker Recognition System

Automatic speaker recognition refers to recognizing persons from their voice. No two individuals sound identical because their vocal tract shapes, larynx sizes, and other parts of their voice production organs are different. In addition to these physical differences, each speaker has his or her characteristic manner of speaking, including the use of a particular accent, rhythm, intonation style, pronunciation pattern, choice of vocabulary and so on. State-of-the-art speaker recognition systems use a number of these features in parallel, attempting to cover these different aspects and employing them in a complementary way to achieve more accurate recognition.

Automatic speaker recognition systems are generally divided into two categories (Figure 1), namely: automatic speaker identification systems which are designed to answer the question “who is the speaker?” or automatic speaker verification systems that aim to answer the question “is the speaker who they claim to be?”.



**Fig. 1.** Structures of ASR system: Identification (Top) and Verification (Bottom)

On other hand, speaker recognition can be classified into text-dependent and text-independent applications. When the same text is used for both training and testing, the system is called to be text-dependent while for text-independent operation, the text used to train and test of the ASR system is completely unconstrained. In contrast, Text-independent speaker recognition usually gives less performance than text-dependent speaker recognition, which requires test input to be the same sentence as training data [10].

## 2.1 Front-End Processing

Front-end processing is the first component in ASR, therefore the quality of the frontend processing will greatly determine the quality of the later other components. Speech signal changes continuously due to the movements of vocal system and it is intrinsically non-stationary. Nonetheless, in short segments, typically 20 to 40 ms, speech could be regarded as pseudo-stationary signal. Speech analysis is generally carried out in frequency domain with short segments and it is often called short-term spectral analysis. The pre-emphasized stream of digital data is analyzed in frames of 20 ms, at intervals of 10 ms. The Hamming window is used to reduce the distortions caused by the discontinuities at the ends of each frame.

Depending on the acoustic front-end of concatenated features, the resulting feature vectors may have from 20 to 50 components. In real-time speaker applications using low-resource devices, like service accessing through portable or embedded device with low storage and computational capabilities, 50-dimensional feature vectors do not seem suitable. For example, for choosing 20 features from 50 original features we have  $4.712 \times 10^{13}$  searches. Therefore, a further feature set reduction is needed.

## 2.2 Acoustic Feature Extraction

The speech waveform contains all information about the speaker, and each step in the extraction process can only reduce the mutual information or leave it unchanged. The objective of the feature extraction is to reduce the dimension of the extracted vectors and thereby reduce the complexity of the system. The main task for the feature extraction process is to pack as much speaker-discriminating information as possible into as few features as possible. The choice of features in any proposed ASR system is of primary concern. Most feature extraction techniques in speaker recognition were originally used in speech recognition. However, the focus in using these techniques was shifted to extract features with high variability among people.

Most commonly used features extraction techniques, such as MFCCS and Linear Prediction Cepstral Coefficients (LPCCs) have been particularly popular for ASR systems in recent years. This transforms give a highly compact representation of the spectral envelope of a sound. Delta-features, regardless of what features they are based, can be computed as a one-to-one function of the features themselves. Therefore, the delta-features do not contain more information than is already in the features, and from the theory, no gain can be achieved by using them together with the features. However, the delta-features can be used as a simplified way of exploiting inter-feature dependencies in sub-optimal schemes.

The number of features should be also relatively low. Traditional statistical models such as the Gaussian mixture model [11] cannot handle high-dimensional data. The number of required training samples for reliable density estimation grows exponentially with the number of features; this problem is known as the curse of dimensionality. The computational savings are also obvious with low-dimensional features. On other hand, dealing with hundreds of features leads to the increase of computational workload of recognition process.

### MFCC Features.

State of the art systems use the Mel Frequency Cepstrum Coefficient for speech and speaker recognition, because they convey not only the frequency distribution identifying sounds, but also the glottal source and the vocal tract shape and length, which are speaker specific features. They are extensions of the cepstral which are used to better represent human auditory models. The MFCCs are calculated as illustrated in Figure 2.

### Differential Features.

Temporal changes, in speech spectra, play an important role in perception. This information is captured in the form of velocity coefficients and acceleration coefficients referred to as differential or dynamic features. The first order derivative of MFCCs is called Delta coefficients and their second order derivative is called Delta-Delta coefficients. The delta coefficients are computed using linear regression:

$$\Delta x(m) = \frac{\sum_{i=1}^j (i)(x(m+1) - x(m))}{2 \times \sum_{i=1}^j i^2} \quad (1)$$

where,  $2j+1$  is the regression window size and  $x$  denotes the cepstrum. The second-order derivatives are computed using the same linear regression applied to a window of delta coefficients.

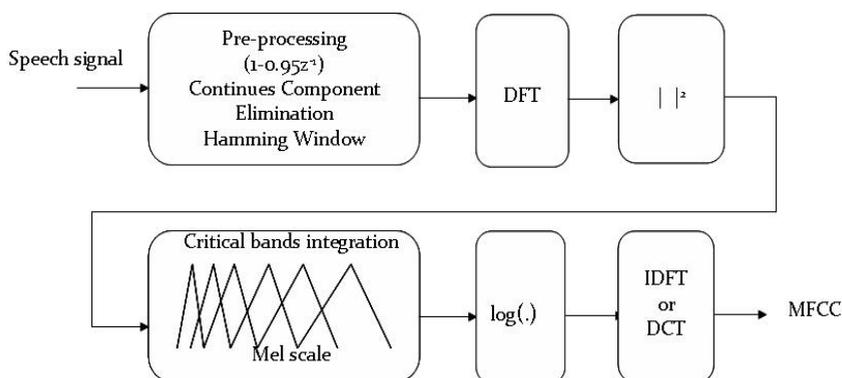


Fig. 2. MFCC features extraction

### 2.3 Classifier

The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified classification result. Recognition process was performed using the KNN classifier.

## 3 Genetic Algorithm

Genetic algorithms [12] are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a single chromosome and apply recombination operators to them so as to preserve critical information. GAs are often viewed as function optimizers, although the range of problems to which GAs have been applied is quite broad. The major reason for GAs popularity in various search and optimization problems is its global perspective, wide spread applicability and inherent parallelism. GA starts with a number of solutions known as population. These solutions are represented using a string coding of fixed length. After evaluating each chromosome using a fitness function and assigning a fitness value, three different operators selection, crossover and mutation- are applied to update the population. The selection is applied on a population and forms a mating pool. Crossover operator is applied next to the strings of mating pool. It picks two strings from the pool at random and exchanges some portion of the strings between them. Mutation operator changes a 1 to 0 and vice versa. An iteration of these three operators is known as a generation. If a stop criterion is not satisfied this process repeats. This stop criterion can be defined as reaching a predefined time limit or number of generations or population convergence. A flowchart of working principles of a simple GA is shown in Fig. 3.

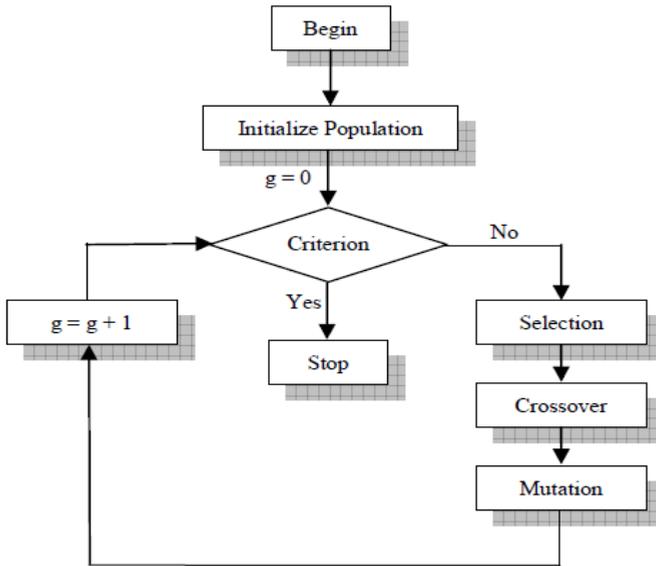


Fig. 3. Simple Genetic Algorithm

### 3.1 GA Optimization Process

Feature selection (or extraction) techniques can be categorized according to a number of criteria. One popular categorization consists of “filter” and “wrapper” to quantify the worth of features [13]. Filters use general characteristics of the training data to evaluate attributes and operate independently of any learning algorithm. Wrappers, on the other hand, evaluate attributes by using accuracy estimates provided by the actual target learning algorithm. Due to the fact that the wrapper model is computationally expensive [14], the filter model is usually a good choice when the number of features becomes very large. In our ASR system, we use an approach similar to one reflected in [15], after pre-processing of speech signals, the front-end is used to transform the input signals into a feature set (feature vector). After that, Feature selection is applied using GA to explore the space of all subsets of given feature set in order to reduce the dimensionality and improve the performance. The feature set optimization process is shown in Fig. 4.

### 3.2 MFCC Features Encoding

For GA-based feature selector, we set the length of chromosomes as the number of features. In a chromosome, each gene  $g_i$  corresponds to the  $i$ th feature. If  $g_i = 1$ , this means we select the  $i$ th feature. Otherwise,  $g_i = 0$ , which means the  $i$ th feature is ignored. By iterations of producing chromosomes for the new generation, crossover and mutation, the algorithm tries to find a chromosome with the smallest number of 1’s and the classifier accuracy is maximized.

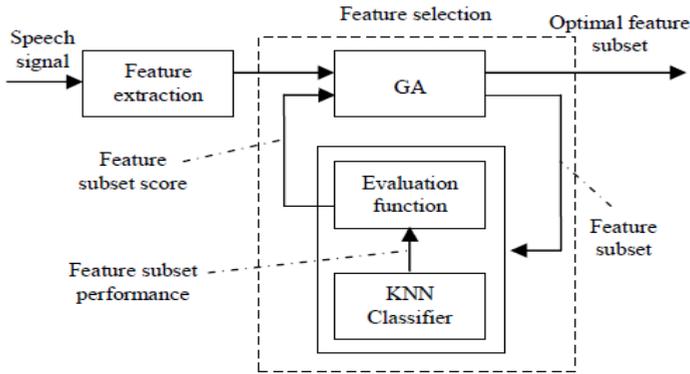


Fig. 4. GA optimization process

### 3.3 MFCC Features Encoding

For GA-based feature selector, we set the length of chromosomes as the number of features. In a chromosome, each gene  $g_i$  corresponds to the  $i$ th feature. If  $g_i = 1$ , this means we select the  $i$ th feature. Otherwise,  $g_i = 0$ , which means the  $i$ th feature is ignored. By iterations of producing chromosomes for the new generation, crossover and mutation, the algorithm tries to find a chromosome with the smallest number of 1's and the classifier accuracy is maximized.

### 3.4 Population Initialization

GA starts by generating an initial population of chromosomes. This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be engendered. Generally, the initial population is generated randomly. The chromosome size is equal to 30 (10MFCC, 10  $\Delta$ MFCC and 10  $\Delta\Delta$ MFCC). We choose the population size  $m = 50$  and the maximum number of iterations  $k = 100$ .

### 3.5 Population Evaluation

The performance criterion is due to Error Rate (ER) and number of feature selected. The best feature subset found is then output as the recommended set of features to be used in the actual design of the classification system. In our experiments, the fitness function is defined according to Equation (2):

$$Fitness = \alpha \cdot \varphi_s + \beta \cdot \frac{|N| - |S|}{|N|} \quad (2)$$

where  $\varphi_s$  is classifier performance for the feature subset  $S$ ,  $|N|$  is the total number of features,  $|S|$  is feature subset length,  $\alpha \in [0;1]$  and  $\beta = 1 - \alpha$ . In our experiment we assume that classification quality is well important as the subset length and we choose  $\alpha = \beta = 0.5$ .

### 3.6 Chromosome Selection

After evaluating all individuals of the population, we apply the elitist selection method. This method allows the genetic algorithm to retain a number of best

individuals for the next generation. These individuals may be lost if they are not selected to reproduce [16].

### 3.7 Crossover

Its fundamental role is to enable the recombination of information contained in the genetic heritage of the population. We applied the one point cross with the variable probability ( $P_{crossover} = \%$  of chromosomes having score  $>$  mean (scores)).

### 3.8 Mutation

A mutation is simply a change of a gene found in a locus randomly determined. The altered gene may cause an increase or a weakening of the solution value that represents the individual ( $P_{mutation} = 0.02$ ).

### 3.9 Replacement

The elitist replacement is the most suitable in our case; it keeps individuals with the best performance from one generation to the next. The weakest individual of the current population is replaced by the fittest individual of the immediately preceding population.

### 3.10 Stop Criterion

As we seek the optimum, we choose our stopping criterion the maximum number of generations even if the optimum is found before, something we can know in advance.

## 4 Results and Discussions

The QSDAS Base [17] is used in this paper. This corpus contains 77 speakers, each speaking 21 sentences partitioned in three sets 10, 10 and 1 respectively. The 77 speakers included in all sets were used during the trials. The effectiveness and performance of our proposed GA-based feature selection algorithm is evaluated using series of experiments. All experiments have been run on Pentium IV, Windows XP, using Matlab 7.0. The classification error rate and feature subset length are the two performance criteria considered. Tables I to IV show the feature vector size reduction and error rate reduction achieved by our genetic algorithm in case of MFCC,  $\Delta$ MFCC,  $\Delta\Delta$ MFCC and MFCC +  $\Delta$ MFCC +  $\Delta\Delta$ MFCC features, respectively.

**Table 1.** ER and Selected Feature Vector Size Reduction (MFCC)

	Size reduction (%)	RR improvement (%)
S <sub>1</sub>	50,00	10,00
S <sub>2</sub>	50,00	10,00
S <sub>3</sub>	40,00	5,00
S <sub>4</sub>	60,00	10,00
S <sub>5</sub>	50,00	10,00

**Table 2.** ER and Selected Feature Vector Size Reduction ( $\Delta$ MFCC)

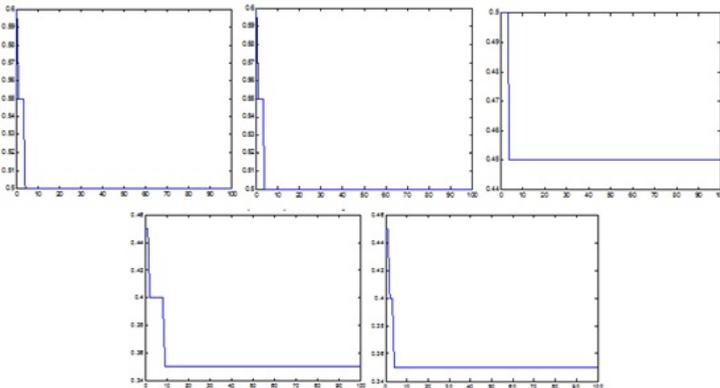
	Size reduction (%)	RR improvement (%)
S <sub>1</sub>	40,00	5,00
S <sub>2</sub>	40,00	10,00
S <sub>3</sub>	50,00	5,00
S <sub>4</sub>	50,00	10,00
S <sub>5</sub>	50,00	10,00

**Table 3.** ER and Selected Feature Vector Size Reduction ( $\Delta\Delta$ MFCC)

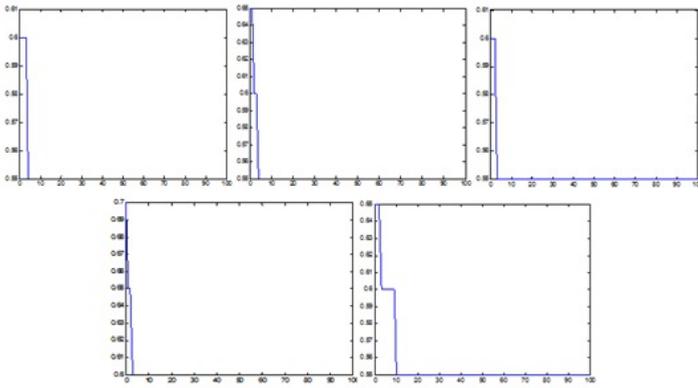
	Size reduction (%)	RR improvement (%)
S <sub>1</sub>	40,00	10,00
S <sub>2</sub>	30,00	10,00
S <sub>3</sub>	40,00	15,00
S <sub>4</sub>	40,00	25,00
S <sub>5</sub>	60,00	10,00

**Table 4.** ER and Selected Feature Vector Size Reduction (MFCC+ $\Delta$ MFCC+ $\Delta\Delta$ MFCC)

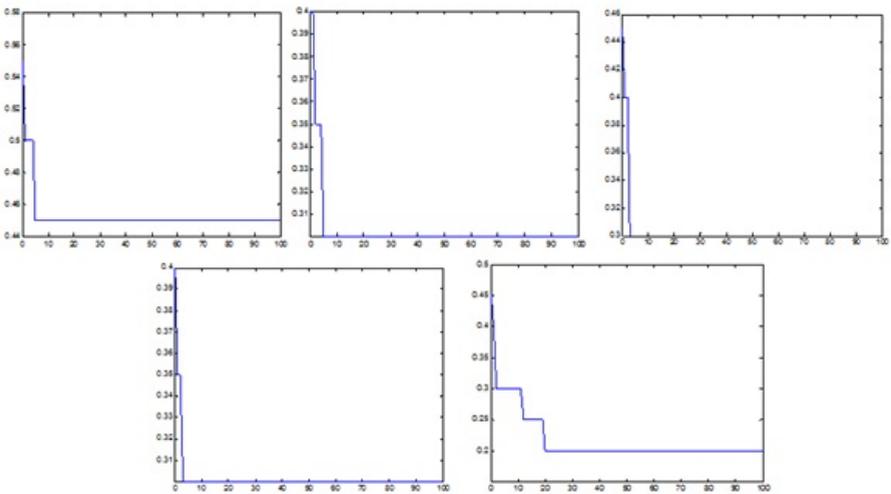
	Size reduction (%)	RR improvement (%)
S <sub>1</sub>	43,33	10,00
S <sub>2</sub>	43,33	15,00
S <sub>3</sub>	40,00	20,00
S <sub>4</sub>	53,33	25,00
S <sub>5</sub>	60,00	25,00



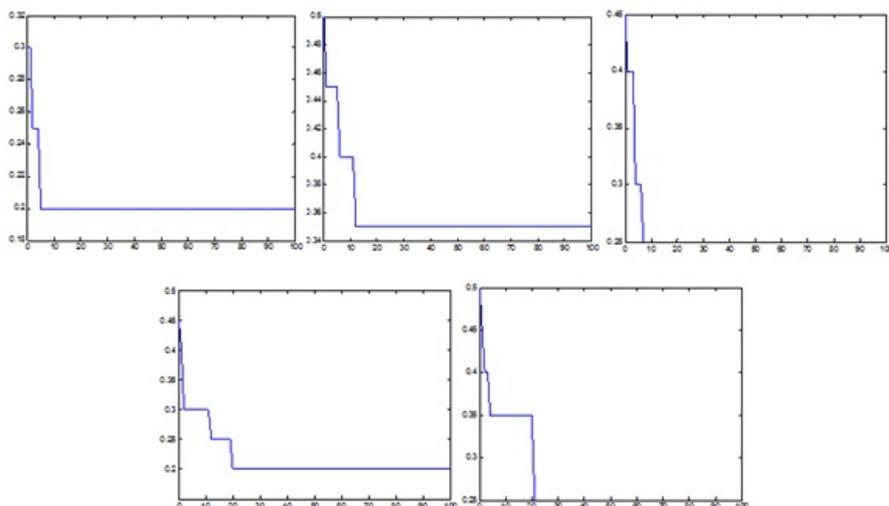
**Fig. 5.** ER using MFCC for the five sets S<sub>1</sub> (top left) to S<sub>5</sub> (bottom right)



**Fig. 6.** ER using  $\Delta$ MFCC for the five sets  $S_1$  (top left) to  $S_5$  (bottom right)



**Fig. 7.** ER using  $\Delta\Delta$ MFCC for the five sets  $S_1$  (top left) to  $S_5$  (bottom right)



**Fig. 8.** ER using MFCC+ $\Delta$ MFCC+ $\Delta\Delta$ MFCC for the five sets  $S_1$  (top left) to  $S_5$  (bottom right)

While Figures 5 to 8 show the evolution of the error rate over the generations. We can see that our GA can reduce the dimensionality of features between 30% (the worst case) and 60% of original features (in the best case). On other hand, our the proposed genetic algorithm has improve the classification rate (reduce the error rate) between 5% (in the worst case) and 25% (in the best case) with better results for  $\Delta\Delta$ MFCC compared to MFCC or  $\Delta$ MFCC. Even using small feature vector, the proposed genetic algorithm can obtain better classification accuracy with smaller. The best result is obtained using a subset selected from a combination of all features vectors (MFCC+ $\Delta$ MFCC+ $\Delta\Delta$ MFCC) which confirm that the inclusion of new parameters improves speaker's discrimination.

## 5 Conclusions and Future Works

In this paper, we addressed the problem of optimizing acoustic feature set by GA-based feature selection algorithm. The GA algorithm adopts classifier performance and the number of the selected features as heuristic information, and selects the optimal feature subset in terms of smallest feature vector size and the best performance of system classifier. The experimental results on QSDAS data sets showed that our GA is able to select the more informative features without losing the performance; the algorithm can obtain better classification accuracy with smaller feature vector which is crucial for real time applications and low resources devices systems. The feature vectors size is reduced over 60% that led to a less complexity of our ASR system and reduce the ER up to 25%. For future works, we prepare another paper on the use of a multi-objective genetic algorithm by separating the two objectives (feature vector size and classification error rate).

## References

1. Day, P., Nandi, A.K.: Robust text-independent speaker verification using genetic programming. *IEEE Transactions on Audio, Speech, and Language Processing* 15(1), 285–295 (2007)
2. Jensen, R.: Combining rough and fuzzy sets for feature selection. Ph.D. thesis, University of Edinburgh (2005)
3. Zamalloa, M., Rodríguez, L.J., Peñarikano, M., Bordel, G., Uribe, J.P.: Comparing genetic algorithms to principal component analysis and linear discriminant analysis in reducing feature dimensionality for speaker recognition. In: *GECCO 2008*, pp. 1153–1154 (2008)
4. Zhao, Q., Zhang, D., Zhang, L., Lu, H.: Evolutionary discriminant feature extraction with application to face recognition. *EURASIP Journal on Advances in Signal Processing*, 1–13 (2009)
5. Harrag, F., El-Qawasmeh, E., Salman Al-Salman, A.M.: Extracting Named Entities from Prophetic Narration Texts (Hadith). In: Zain, J.M., Wan Mohd, W.M.b., El-Qawasmeh, E. (eds.) *ICSECS 2011, Part II. Communications in Computer and Information Science*, vol. 180, pp. 289–297. Springer, Heidelberg (2011)
6. Rajavarman, V.N., Rajagopalan, S.P.: Feature Selection in Data-Mining for Genetics Using Genetic Algorithm. *Journal of Computer Science* 3(9), 723–726 (2007)
7. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Sciences* 180(14), 2633–2634 (2010)
8. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. *Neurocomputing* 75(1), 61–63 (2012)
9. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10(5), 335–347 (1989)
10. Nemati, S., et al.: Text-independent speaker verification using ant colony optimization-based selected features. *Expert Systems with Applications* 38, 620–630 (2011)
11. Reynolds, D.A., Rose, R.C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Transactions on Speech and Audio Processing* 3(1), 72–83 (1995)
12. Mitchell, M.: An introduction to genetic algorithms. A Bradford book. The MITpress, Cambridge (1999) (fifth printing)
13. Chandra, E., Nandhini, K.: Learning and Optimizing the Features with Genetic Algorithms. *International Journal of Computer Applications* 9(6), 1–5 (2010)
14. Langley, P.: Selection of Relevant Features in Machine Learning. In: *Proceedings AAAI Fall Symposium Relevance* (1994)
15. Casillas, J., Cordon, O., Del Jesus, M.J., Herrera, F.: Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. *Information Sciences Journal* 136, 135–157 (2001)
16. Rokach, L.: Genetic Algorithm-based Feature Set Partitioning for Classification Problems. *Journal Pattern Recognition* 41(5), 1676–1700 (2008)
17. Harrag, A., Mohamadi, T.: QSDAS: New Quranic Speech Database for Arabic Speaker Recognition. *AJSE Journal* 35(2C), 7–19 (2010)