Optimization of hidden Markov model with Gaussian mixture densities for Arabic speech recognition

ABDELMADJID BENMACHICHE , AMINA MAKHLOUF
Department of Computer Science
Chadli Bendjedid, University, El-Tarf, Algeria, PB 73, 36000.
Laboratory LRI, Badji Mokhtar University, Annaba, Algeria, PB 12. 23000 E-mail:
benmachiche@hotail.fr or: benmachiche-abdelmadjid@univ-eltarf.dz

Abstract: - Speech recognition applications are becoming more and more useful nowadays. In automatic speech recognition (ASR) systems, hidden Markov models (HMMs) have been widely used for modeling the temporal speech signal. Iterative algorithms such as Forward - Backward or Baum-Welch are commonly used to locally optimize HMM parameters (i.e., observation and transition probabilities). In this paper we propose a general approach based on Genetic Algorithms (GAs) to evolve HMM with Gaussian mixture densities. The problem appears when experts assign probability values for HMM, they use only some limited inputs. The assigned probability values might not be accurate to serve in other cases related to the same domain. We introduce an approach based on GAs to find out the suitable probability values for the HMM to be mostly correct in more cases than what have been used to assign the probability values. For this purpose, a sample database containing speech files of Algerian speakers is used.

Key-Words: - Automatic speech recognition - Acoustic information - Genetic algorithm (GA) - GA/HMM - Gaussian mixture densities - Baum-Welch.

1 Introduction

In our minds the aim of interaction between a machine and a human is to use the most natural way of expressing ourselves, through our speech. The performance of the ASR systems relies on conventional hidden Markov models (HMMs) which are based on maximum likelihood estimation (MLE) techniques.

In order to improve the performance of ASR systems, researchers have explored other paradigms like neural and Bayesian networks, discriminative training techniques, state duration modeling and the use of support vector machines with HMM.

The earliest attempts to devise systems for automatic speech recognition by machine were made in the 1950's, when various researchers tried to exploit the fundamental ideas of acoustic-phonetics. In 1952, at Bell Laboratories, Davis, Biddulph, and Balashek built a system relied heavily on measuring spectral resonances during the vowel region of each digit. In 1970's speech recognition research achieved a number of significant milestones. First the area of isolated word or discrete utterance recognition became a viable and usable technology based on the fundamental studies by Velichko and Zagoruyko in Russia [1], Sakoe and Chiba in Japan [2] and Itakura in 1975 in the United States [3].

In recognizing syllables or isolated words, the human auditory systems perform above chance level already at -18dB signal-to-noise ratio (SNR) and significantly above it at -9dB SNR. No ASR system is able to achieve performance close to that of human auditory systems in recognizing isolated words or phonemes under severe noisy conditions, as has been confirmed recently in an extensive study by Sroka in 2005[4].

In order to improve the performance of ASR systems, researchers have explored other paradigms like neural and Bayesian networks, discriminative training techniques, state duration modeling and the use of support vector machines with HMM [5].

In 2008, Norris the author present a Bayesian model of continuous speech recognition [6]. It is based on Shortlist and shares many of its key assumptions: parallel competitive evaluation of multiple lexical hypotheses, phonologically abstract prelexical and lexical representations, feed forward architecture with no online feedback, and a lexical segmentation algorithm based on the viability of chunks of the input as possible words.

The work of Graves in 2013 investigates deep recurrent neural networks (RNNs) [7], which combine the multiple levels of representation, that have proved so effective in deep networks with the flexible use of long range context that empowers RNNs. When trained end-to-end with suitable regularisation, they find that deep Long Short-term Memory RNNs achieve a test set error of 17.7% on the TIMIT phoneme recognition benchmark

The language used in this work is Arabic. It is known that, Arabic is the fifth most widely spoken language in the world, with more than 300 million native speakers, cutting across a wide geographical area from North Africa to the Middle East. Also, it is one of the six official languages adopted in the United Nations. It is the official language in some twenty-two countries, whereas there are substantial Arabic-speaking communities in many countries. In addition, it is the liturgical and worship language for more than one billion and half Muslims worldwide [8].

In addition, Arabic speech recognition faces many challenges. For example, Arabic has short vowels which are usually ignored in text. Therefore, more confusion will be added to the ASR decoder. Additionally, Arabic has many dialects where words are pronounced differently. The work in [9] summarized the main problems in Arabic speech recognition, which include Arabic phonetics, diacritization problem, grapheme-to-phoneme relation, and morphological complexity. Bourouba in 2006 presented a new HMM/support vectors machine (SVM) (k-nearest neighbor) for recognition of isolated spoken words [10]. Sagheer in 2005 propose a novel visual speech features representation system. They used it to comprise a complete lip-reading system [11].

In addition, the work of Muhammad in 2011 evaluated conventional ASR system for six different types of voice disorder patients speaking Arabic digits [12]. Mel-frequency cepstral coefficients (MFCCs) and Gaussian mixture model (GMM)/HMM are used as features and classifier, respectively. Recognition result is analyzed for types of diseases.

At AT&T Bell Labs, began a series of independent Pazhayaveetil et al in 2007[13]. They used a wide range of sophisticated clustering algorithms to determine the number of distinct patterns required to represent all variations of different words across a wide user population. In 1980’s a shift in technology from template based approaches to statistical modeling methods, especially the hidden Markov model approach.

In this paper, we investigate the effectiveness of using GA for optimizing the structure and parameter learning of HMM. A GA is a robust general purpose optimization technique that evolves a population of solutions [14]. It is easy to hybridize other algorithms such as Baum–Welch training within a GA. Furthermore, it is possible to design operators that favour biologically plausible changes to the structure of an HMM. That is, to ensure that modules of the states are kept intact. The main point of this work is based on the quality of modelization of data (called observations) made by HMM. Our goal is to propose algorithms that improve this quality. The criterion used to quantify the quality of HMM is the probability that a given model generates a given observation. To solve this problem, we use as we have already mentioned a genetic hybridization of HMM and we propose representation methods of a gene and the steps method for fitness evaluation of populations of each created generation by GA. We prove that GA outperforms the traditional Baum-Welch in the estimation of the parameters, resulting in higher accuracy in the state decoding. In our experiments, the GA achieved 87.3% accuracy while the EM 74%.

This paper is organized as follows. Section 2, we will deal with the background knowledge to understand our proposed approach, and we will explain all the method used in this work. In Section 3, we present some experimental results produced by the proposed method. Section 4 wraps up the paper with a conclusion and perspectives.

2 GA/HMM based ASR system

In this section we present the structure of an ASR system. This system was divided into three modules according to their role. The first module is feature extraction to capture the most relevant and discriminate characteristics of the signal to recognize. The second subsystem is training module, whose function is to create the knowledge about the speech and language to be used in the system. Final module is the recognition module, whose function is tried to figure out the meaning of the input speech given in the testing phase.

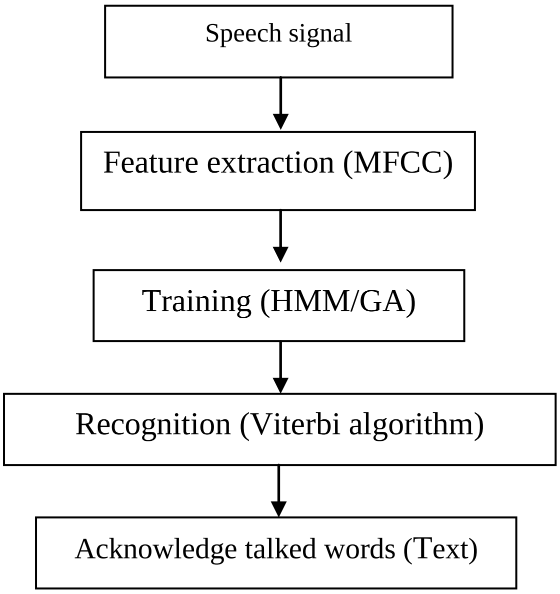

Speech signal
Feature extraction (MFCC)
Training (HMM/GA)
Recognition (Viterbi algorithm)
Acknowledge talked words (Text)

Fig1. Architecture of our ASR system.

2.1 Pre-processing and feature extarction

Audio feature extraction begins by using preprocessing techniques such as signal enhancement and environment sniffing to help prepare the incoming audio stream for the feature extraction step [15]. Generally, the audio is broken down into sliding window “frames” and features extracted from each frame. In many cases, the frame size is on the order of 10ms. Many results have been reported in the literature regarding the extraction of audio features for clean and noisy speech conditions. MFCCs [16] and linear prediction coefficients (LPCs) [17] represent the most commonly used acoustic features. Additional research is ongoing in the field of noise robust acoustic features. After acoustic feature extraction, first and second derivatives of the data are usually concatenated with the original data to form the final feature vector. The original data is also known as the static coefficients while the first and second derivatives are also known as delta and delta-delta or acceleration coefficients.

Most speech parameter estimation techniques are easily influenced by the frequency response of the communication channel. For this reason, the work in [18] has developed a technique that is more robust to such steady-state spectral factors in speech. The approach is conceptually simple and computationally efficient.

The proposed method for speech analysis is based on RelAtive SpecTral Analysis-Perceptual Linear Predictive method (RASTA-PLP) for feature extraction. This technique is an improvement of the traditional PLP method. It consists in a special filtering of the different frequency channels of a PLP analyzer. The previous filtering is done to make speech analysis less sensitive to the slowly changing or steady-state factors in speech. The RASTA method replaces the conventional critical-band short-term spectrum in PLP and introduces a less sensitive spectral estimation. For the RASTA-PLP features, an additional filtering is applied after decomposition of the spectrum into critical bands. This RASTA filter suppresses the low modulation frequencies which are supposed to stem from channel effects rather than from speech characteristics.

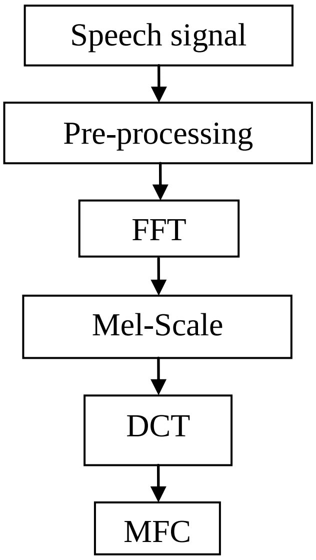
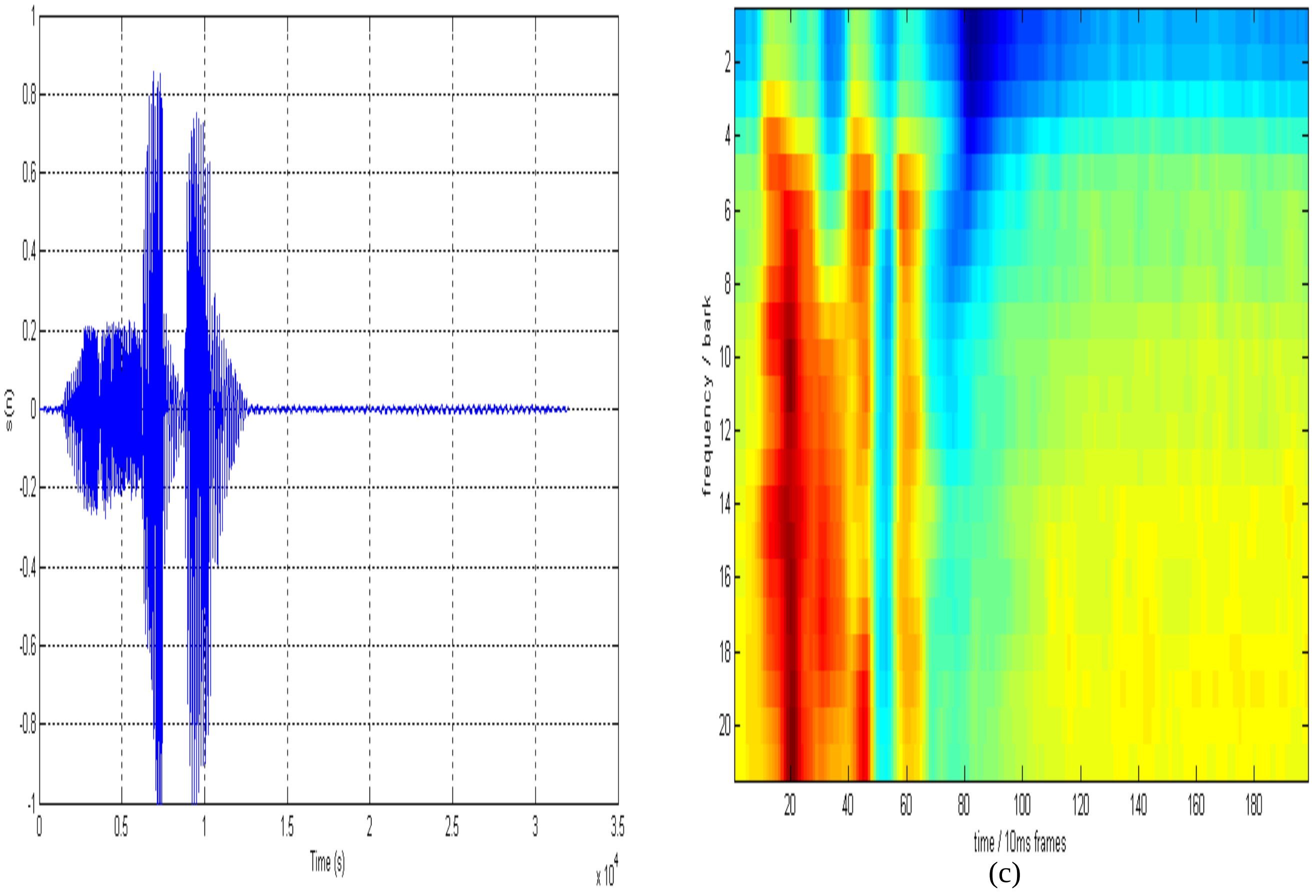


Fig 2. MFCC extraction.

The low cut-off frequency of the filter determines the fastest spectral change of the log spectrum. The high cut-off frequency indicates the fastest spectral change which is preserved in the output parameters. Convolution noise is attenuated by the higher values of the band-pass filter. This means that the current analysis result depends on previous outputs stored in the memory of the recursive RASTA filter. In this sense, the analysis results depend on the time in where the analysis starts. The PLP and RASTA-PLP streams were augmented by their delta features.

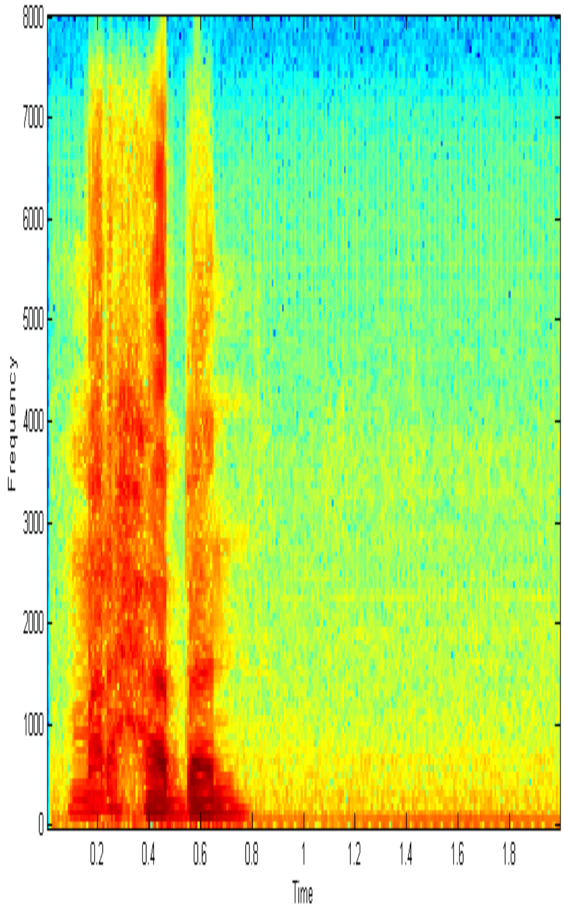
In this work, the sampled speech data are segmented into 0.025 seconds frames with a 0.010 seconds overlapping of two consecutive windows. After application of a hamming window, each of these frames is analyzed using the RASTA-PLP speech analysis technique, in order to make speech analysis more robust to spectral.

Figure 3 shows an example of the RASTA-PLP features.



(a)

(c)



(b)

Fig 3. Example of a speech signal of the Arabic word “/marhaban/” (a), its spectrogram (b) and the set of RASTA-PLP spectral features (c).

2.2. Research method

The well-known Baum-Welch algorithm can be used to simultaneously estimate the state transition probabilities and the observation probabilities in a maximum likelihood framework from the sequence of observations. The states in HMM are assume as discrete values. However, the observations can have either discrete (limited) values, modelled with probabilities in matrix B, or continuous (or continuous discretised), modelled by conditional probability density functions.

In this paper, GA was applied to optimize the Baum-Welch algorithm in left-to-right HMM. The result between HMM system and hybrid GA/HMM was compared to analyze how GA can improve the rate of recognition in hybrid GA/HMM.

2.2.1. Hidden Markov Model (HMM)

Speech recognition systems generally assume that the speech signal is a realization of some message encoded as a sequence of one or more symbols. To perform the reverse operation of recognizing the underlying symbol sequence given a spoken utterance.

In our work, we used precise statistical models: the Hidden Markov Models (HMM) which have emerged as the predominant technology in speech recognition in recent years. These models have proved to be better adapted to the problems of speech recognition.

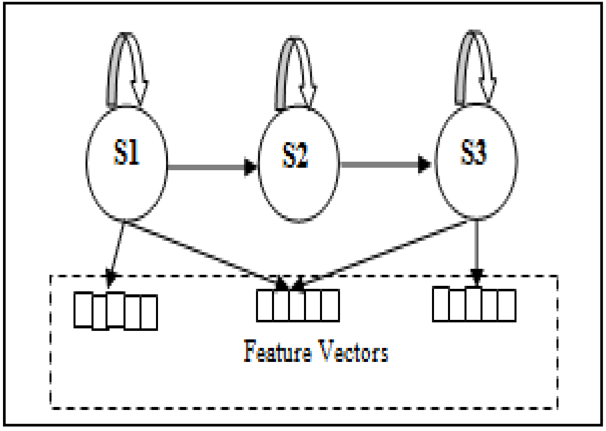


Fig 4. Structure of HMM.

The HMMs are a class of models in which the distribution that generates an observation depends on the state of an underlying but unobserved Markov process [19]. Thus a HMM is a combination of two processes, namely a Markov chain which determines the state at time t, St=st, and a state-dependent process which generates the observation Xt = xt depending on the current state st. In most cases a different distribution is imposed for each possible state of the state space. A hidden Markov Model is characterized by:

- S: the set of states
- O: the sequence of observations, each of which is drawn from a vocabulary, <o_1, o_2, …, o_T>
- N is the number of states in the model.
- M is the number of mixtures in the random function.
- π ={π_i}: initial state distribution
  $\pi_{㋓} = P(q_1 = i), \quad 1 \leq i \leq N$
  (1)
- A={a_{ij}}: the transitional probability matrix

$a_{ij} = P(q_{t+1} = j|q_t = i), \quad 1 \leq i,j \leq N,\ 1 \leq t \leq T$ (2)

- B={$b_j(o)$}: the probability distribution at each state, where $b_i(o_t)$ is the probability of observation o_t generated from state i

The HMM outputs are modeled as continuous density output probabilities by representing each output probability as a Gaussian mixture. Thus the output probability for an HMM in sate j is given by:

$$b_j(o) = \sum_{k=1}^{M} w_{jk} N(o, \mu_{jk}, \sigma_{jk}^2) \qquad (3)$$

Here the sum is over the mixtures (k), where $N(o, \mu_{jk}, \sigma_{jk}^2)$ denotes a Gaussian distribution for observation vector o with mean vector μ_{jk} and variance matrix σ_{jk}^2 and w_{jk} are the mixture weights for each state.

As the probability distribution for the output probabilities must integrate to one, as in Eq. (4):

$$\sum_{k=1}^{M} w_{jk} = 1 \qquad (4)$$

There are three fundamental problems related to the HMM; the evaluation problem, the problem of determination of the path of states, the problem of learning (supervised or unsupervised). Some other problems related to HMM are: the overflow of representation of numbers in machine, insufficient data for learning, the update of models when processes vary in time, the choice of HMM architecture the best adapted to data, the choice of a good initial estimate of the probabilities of the HMM. The Forward algorithm allows the calculation of the likelihood of the observations, the Viterbi algorithm finds the optimal path which is the most likely to follow and the Baum-Welch algorithm performs the supervised learning (re-estimate the parameters of HMM).

2.2.2. Genetic algorithm (GA)

The GA represents an advanced numerical search and heuristic optimization method inspired by the biological theory of evolution [14].

The first one who described the genetic algorithms is John Holland which is in the 1960s and was subsequently studied by Holland and coworkers at the University of Michigan in the 1960s and 1970s [14]. Pioneered by John Holland, they attempt to mimic natural selection by using a population of competing solutions that evolve over a series of generations.

These problems are very common in predicting phenomena of real-world, machine modeling, machine learning, and optimization.

In general, GA is initialized with little knowledge about the given matter to be solved and a searching process is performed in parallel for a complex and vast search space.

Generally, a simple GA cycle consists of four operations: Fitness evaluation, selection, genetic operations, and replacement. In a simple GA cycle, there exists a population pool of chromosomes. The chromosomes are the encoded form of the potential solutions and all GA operations except the fitness evaluation to be performed with this form of solutions.

Initially, the population is generated randomly and the fitness values of all the chromosomes are evaluated by calculating the objective function in the decoded form of chromosomes. After the initialization of the population pool, the GA evolution cycle is begun. At the beginning of each generation, the mating pool is formed by selecting some chromosomes from the population. This pool of chromosomes is used as the parents for the genetic operations to generate the offspring or the subpopulation. The fitness values of the offspring are also evaluated. At the end of the generation, some chromosomes in the population will be replaced by the offspring.

The above generation is repeated until the termination criterion is met. By emulating the natural selection and genetic operations, this process will hopefully leave the best chromosomes or the highly optimized solutions to the problem in the final population.

2.2.3. The proposed GA/HMM training

In this paper, we explore the use of GAs for evolving HMMs in the training stage. The genetic algorithm will manipulate individuals who are going to be candidate solutions to the problem that we want to resolve. Naturally, in the problem of learning of the HMM, the individuals are HMM. We must now encode the HMM in chromosomes on which will apply the genetic operators.

As we have seen before, the problem could be stated as: How to calculate these three parameters? In the first case, using an EM algorithm, we calculate w, μ and Σ, but some problems appear [20]:

- This algorithm is very sensitive to the initialisation of the parameters.
- We always get a local optimum.
- The algorithm does not use the ground truth data.

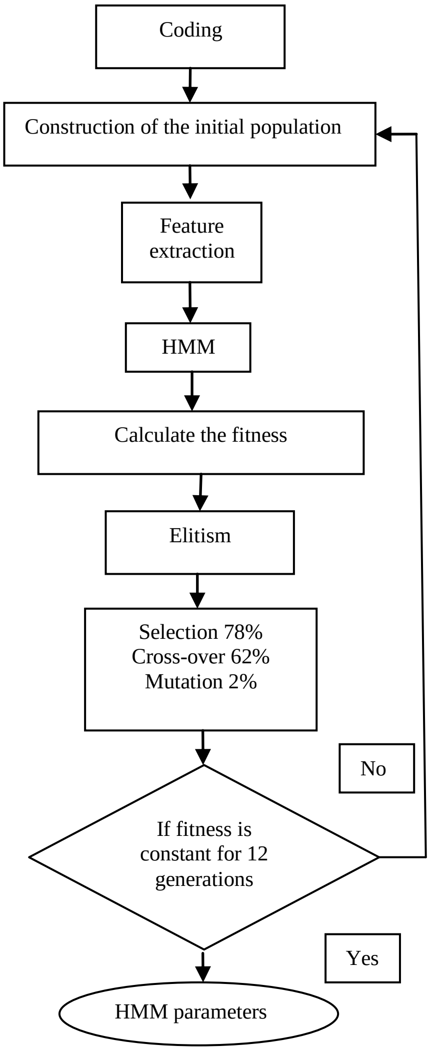


Fig 5. The proposed hybrid GA/HMM.

In the second case, we use a GA to calculate the w, μ and Σ parameters. This algorithm is not so very sensitive to the initialisation of the parameters, because GA begins with many individuals and mutation operator generates new individuals in different search zones. This characteristic allows to the GA to avoid getting the local optimum and hopefully finding a global optimum. Moreover, during the training process we know exactly how to match the output of the algorithm with the ground truth and use this as the fitness function.

We give below the frame of this algorithm which is inspired from [21] which will seek to obtain optimal HMM. We use a marker named "parent" simply to treat only the necessary individuals during the optimization and the evaluation.

1) Initialization: Create a population of size S, the most natural encoding is to fabricate the chromosome by reorganizing all the coefficients of

the HMM. The simplest way is to juxtapose all rows of all matrices. We so obtain a coding in real numbers while respecting the constraints related to the HMM. The representation of a population is defined by the Fig. 6 as follows:

M	π_1	…	π_N	$a_{1,1}$	…	$a_{1,N}$	$a_{2,1}$	…	$a_{N,N}$	
w_1	w_M	μ_1	…	$\mu_{M,T}$	$\sigma^2_{1,M}$	…	$\sigma^2_{M,T}$			

Fig 6. Chromosome representation method in the GA/HMM training

The set of parameters values: weights, means and variances (w, μ, σ2) of the observation distributions (the B "matrix"), are first initialized using a segmental k-means procedure, the k-means is used to cluster the vectors in each state into a set of M clusters (using a Euclidean distortion metric and a vector quantization (VQ) design algorithm). From the clustering, an updated set of model parameters is derived as follows:

1- $\hat{w}_{jk}$ = number of vectors classified in cluster k of the *j*th state/number of vectors in state j;

2- $\hat{\mu}_{jk\,d}$ = *d*th component of mean of vectors classified in cluster *k* of state *j*;

3- $\hat{\Sigma}_{jk\,rs}$ = (*r*, *s*)th component of covariance matrix of vectors classified in cluster *k* of State *j*.

For the other chromosomes are randomly created.
No individual is marked "parent." Read an observation O.
2) Optimization: Apply to each HMM of the unmarked population "parent" the Baum-Welch algorithm from the observation O.

3) Evaluation: The quality of an individual (also called fitness) describes the adequacy of this one with its environment. More precisely, the fitness function is an evaluation mechanism of the chromosome; a higher fitness value reflects the chances of the chromosome to be chosen in the next generation. In the problem of optimizing an HMM, it is desired to quantify the ability of an HMM to learning an observation. In our GA, the fitness values are the results of the objective function. The likelihood $P(o_j|\lambda_i)$ is an appropriate criterion used in the objective function to determine the quality of the chromosomes. As mentioned previously, the Baum-Welch algorithm has to maximize the likelihood probability such that a given HMM λ_i generated the training utterances o_i:

$$f(\ddot{e}_i) = \frac{P_n}{\sum_{i=1}^{N} P_i} \qquad (5)$$

The average probability P_n is therefore given as follows:

$$P_n = \frac{\sum_{i=1}^{M} log\left(P(o_i|\lambda_i)\right)}{M} \qquad (6)$$

Where, P_n is the average probability of the λ_i model, N is the number of individuals in a population and M representing the number of rows in o_i.

It is proven that Baum-Welch algorithm leads to a local maximum of function$f(\ddot{e}_i)$. However, it is possible that other better maxima of $f(\ddot{e}_i)$exist for given training set. In this paper we tried to overcome this problem by using genetic algorithms for maximization of $f(\ddot{e}_i)$ [22].

4) Selection: Among all the individuals of the population, select a number S'<S, which will be used as parents to regenerate the S-S' other individuals not selected. The selection is done according to the best calculated scores in step 3. Each selected individual is marked "parent."

5) Crossover/recombination: For each unmarked individual "parent" randomly select two individuals from the population of those marked "parent" and cross them to form two offsprings. The crossover probability pc determines the number of offspring individuals H (H= p_cK). In this work we use the single-point crossover for more details).The crossover used in this work is a crossover point, and is realized between two rows of the matrices of the HMM. This allows obtaining in return two children. It retains only one of both children, at random.

6) Mutation / normalization: On each unmarked individual "parent" we apply the mutation operator. Offspring produced by crossover cannot contain information that is not already in the population, so an additional operator, mutation, is required. Mutation generates an offspring by randomly changing the values of genes at one or more gene

positions of a selected chromosome. Each coefficient is modified according to the value of the mutation probability. After treating an individual, we apply on him an operator of normalization, to ensure that this individual still answers the constraints of the HMM. We should verify that the matrices of the HMM are stochastic. This operator is applied after the operation of mutation because subsequent operations imperatively work on HMM.

7) Evaluation of the stop condition: If the maximum number of iterations is not reached, then return to step 3, otherwise go to step 8.

8) Finally return the best HMM among the current population.

Note that this algorithm has been adapted to optimization of vectors of observations. The re-estimation made so that the HMM has a maximal probability to generate the set of vectors.

2.3. Recognition

Recognition is done by a discriminant model. That is to say that learning will be associated with each word learned an HMM. Recognition will be done by calculating for each known HMM its probability of generating the word to recognize. We will recognize the word which the associated HMM obtained a maximum score.

Decision stage chooses the HMM who has the highest probability of generating the input data. In our work the decision is performed using Viterbi algorithm.

The idea of the Viterbi algorithm is used to find the best state sequence, $q_1^*, q_2^*, \dots, q_T^*$ given the observation sequence $O_1, O_2, \dots, O_T$. The highest probability along a single path, which accounts for the first k observations and ends in Si at time k, is defined as:

$$\delta_k(i) = \max_{q_1,q_2,\dots,q_{k-1}} P[q_1 q_2 \dots q_k = S_i, O_1, O_2, \dots, O_k | \lambda] \qquad (7)$$

It can be induced that:

$$\delta_{k+1}(j) = \max_i \ [a_{ij}\,\delta_k(i)_1].\, b_j(O_{k+1} \qquad (8)$$

The best state sequence can be retrieved by keeping track of the argument that maximizes previous equation for each k and j. The observation probability distribution $b_j(O_k)$ is a Gaussian mixture likelihood function as mentioned above.

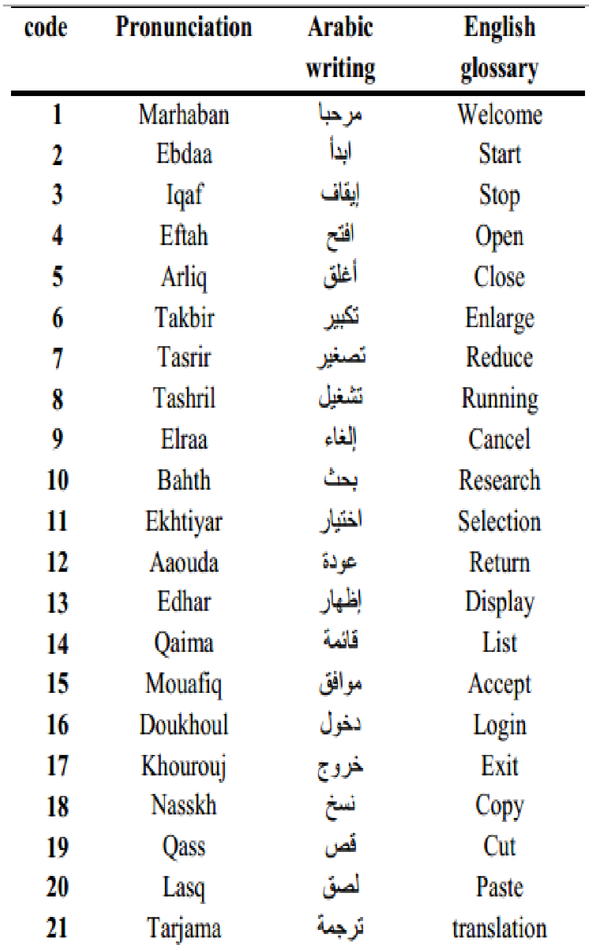
3 Experimental results and discussion

3.1. Database

In this first work, a multi-speakers database was built for a speech recognition task, this database was recorded in a real environment (very noisy classroom), it contains pronunciations of Arabic words isolated, taken at a sampling frequency of 16 KHz.

The database contain a dictionary of 25 Arabic commands, is collected from 18 individual speakers (2 female and 16 male), these speakers are from different regional dialects, and each speaker pronounces each word 9 times with different modes of pronunciation (normal, slow and fast).

The distance between the microphone and the speaker is adjustable to add diversification in the audio streams during the learning, the average distance is to 14.5 cm. In our basic corpus which contains only isolated words, the size of each record is 2 seconds which is enough time to utter a word slowly in Arabic.



code	Pronunciation	Arabic writing	English glossary
1	Marhaban	مرحبا	Welcome
2	Ebdaa	ابدأ	Start
3	Iqaf	إيقاف	Stop
4	Eftah	افتح	Open
5	Arliq	أغلق	Close
6	Takbir	تكبير	Enlarge
7	Tasrir	تصغير	Reduce
8	Tashril	تشغيل	Running
9	Elraa	إلغاء	Cancel
10	Bahth	بحث	Research
11	Ekhtiyar	اختيار	Selection
12	Aaouda	عودة	Return
13	Edhar	إظهار	Display
14	Qaima	قائمة	List
15	Mouafiq	موافق	Accept
16	Doukhoul	دخول	Login
17	Khourouj	خروج	Exit
18	Nasskh	نسخ	Copy
19	Qass	قص	Cut
20	Lasq	لصق	Paste
21	Tarjama	ترجمة	translation

Fig 7. Our proposed corpus of Arabic commands

3.2 Experimental results

As a standard procedure in evaluating machine learning techniques, the dataset is split into a training set and a test set. In our work we have used ⅔ of the data for the learning stage and the remaining ⅓ to test the effectiveness of our ASR system.

An ASR system using RASTA-PLP method as audio features, and a GA/HMM for the speech modeling, was implemented as described in the previous sections. The whole programming was implemented in matlab; As a result, and by integrating the first and second derivative of the parameters, we obtain matrix of 27 parameters. The GA/HMMs recognizers were built using the HTK (Hidden Markov Model Toolkit) [23].

In order to evaluate the performance of the proposed system, various kinds of instance with different GA control parameters have been solved with our algorithm. We ran each instance 15 times with a different number of mixture, a different crossover probability values between 0.4-0.9, and we kept the value 0.01 for the mutation probability, also we take a maximum number of iteration for EM algorithm equal to 40.

After several tests over the training data and the testing data, we found that the fitness function in GA is better with a 180 iterations which yields to better results for optimizing HMM parameters. The figure also show that the GA fitness function increase faster especially in the beginning iterations.

In the following, we present our experimental results on our ASR system over a range of noise levels using these two models. Artificial white Gaussian noise was added to simulate various noise conditions.

The experiment was conducted under a mismatched condition the recognizers were trained at 20dB SNR, and acoustic white Gaussian noise, ranging from -5dB to 20dB in steps of 5 dB SNR, is added to the test data.

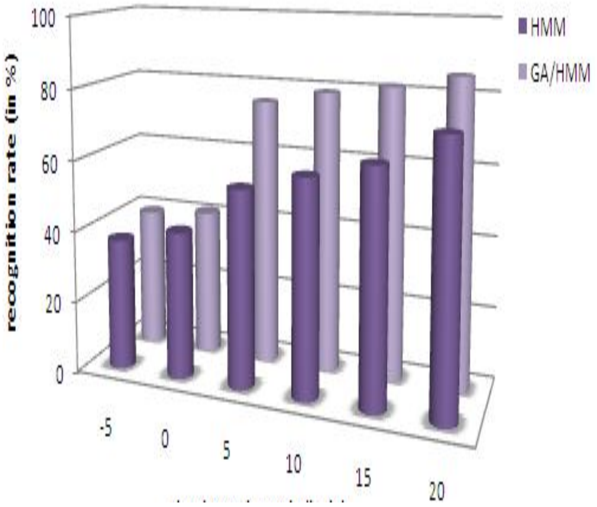


Fig 8. Recognition rates of ASR system for varying SNR levels

We evaluate and compare the systems performance by calculating the average recognition rate for each of the SNR levels, over all utterances for each word. Moreover, HMM model is trained by using the traditional method (Baum- Welch algorithm). Based on Fig.5, we can see that in most of cases the recognition rates obtained with our GA/HMM system are better compared to those obtained with the HMM-based system, and the percentage increase amounting from 3.2% to 16.3%, along with the increase in the size of the population.

4 Conclusion and perspectives

This paper presents an approach for an ASR system using a HMM with a Gaussian mixture densities for modeling the Arabic speech. We present the use of the GA to further optimize the solution found by the Baum-Welch algorithm.

From the several test results, we conclude that the system modeled by HMM trained by our GA/HMM training have higher rates of recognition than the HMM trained by the Baum-Welch algorithm.

Moreover, the result of the classification by using Baum-Welch is greatly dependant on the initialization of the parameters in contrast to the GA that give stable results independently of the initial values.

For future work, we are planning to cover more issues about improving the performance of the system, we will try to enhance the size of our database and increase the number of speakers, and we also intend to test our system with other alternatives methods of fusion.

References:

[1] VELICHKO, V. M., & ZAGORUYKO, N. G. (1970). *Automatic Recognition of 200 words, International journal of Man-Machine Studies*, 223(2).

[2] SAKOE, H., & CHIBA, S. (1978). *Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustic,Speech, Signal Processing., ASSP*-26 (1), 43–49.

[3] F. ITAKURA, *Minimum prediction residual applied to speech recognition,* IEEE Transactions on Acoustics,Speech, Signal Processing, ASSP-23(1), pp. 67-72. 1975.

[4] SROKA, J.& BRAIDA, L. (2005). *Human and Machine Consonant Recognition*. *Speech Comm.* , 45(4), 401–423.

[5] ZHI-YI, Q. YU, L., LI-HONG, Z., & MING-XIN, S. (2006). *Hybrid SVM/HMM architectures for speech recognition*. *Proceedings of the First International Conference on Innovative Computing, Information and Control*, *2*, 100–104.

[6] NORRIS, D., & MCQUEEN, J.M. (2008). *Shortlist B: A Bayesian model of continuous speech recognition. Psychological Review,* 115(2), 357-395.

[7] GRAVES. A., MOHAMED, A.-R, & HINTON, G. (2013). *Speech recognition with deep recurrent neural networks*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Vancouver, British Columbia, Canada, 6645–6649.

[8] KETTANI, H. (2008). *2010 world Muslim population, The 8$_{th}$ Hawaii International Conference on Arts and Humanities*, Honolulu.

[9] ELMAHDY, M., & GRUHN, R. (2009). *Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition. In proceedings of the 8th international symposium on natural language*.

[10] H. BOUROUBA, R. DJEMILI, M. BEDDA, AND C. SNANI, *New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition*, 2nd Information and Communication Technology (ICTTA'06), pp. 1264 – 1269. 2006.

[11] SAGHEER, A., TSURUTA, N., TANIGUCHI, R.I., & MAEDA, S. (2005). *Hyper column model vs. fast DCT for feature extraction in visual Arabic speech recognition. In: Proceedings of the fifth IEEE international symposium on signal processing and information technology*, 761 – 766.

[12] MUHAMMAD, G., ALMALKI, K., MESALLAM, T., & FARAHAT, M. (2011). *Automatic Arabic digit speech recognition and formant analysis for voicing disordered people*. *IEEE symposium on computers and informatics (ISCI),* pp. 699–702.

[13] PAZHAYAVEETIL, U.C. (2007). *Hardware implementation of a low power speech recognition system. PhD. dissertation, Dept. Elect. Eng.,* North Carolina State Univ., Raleigh, NC.

[14] A. BENMACHICHE, T. BOUHADADA, M, T. LASKRI, A. ZENDI, *A dynamic navigation for autonomous mobiles robots,* Intelligent Decision Technologies, ISSN 1875-8843 (E), 10 (1). 2016.

[15] RABINER, L., & JUANG, B.-H. (1993). *Fundamentals of speech recognition*. *Prentice Hall*, New Jersey, ISBN 0- 13-015157-2.

[16] DAVIS, S.B. & MERMELSTEIN, P. (1990). *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. *Published in Book of readings in speech recognition*, (pp. 65–74). Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[17] B. S. ATAL, AND S. L. HANAUER, "*Speech analysis and synthesis by linear prediction of the speech wave*," Journal of the Acoustical Society of America, Vol. 50, No. 2, pp. 637-655. 1971.

[18] HERMANSKY, H., MORGAN, N., BAYYA, A., & KOHN, P. (1991). *RASTA-PLP Speech Analisys. ICSI Technical Report TR-91-069*, Berkeley, California.

[19] EPHRAIM, Y. AND MERHAV, N. (2002). *Hidden Markov Processes*, *IEEE Transactions on Information Theory*, 48(6).

[20] PEREZ, Ó, PICCARDI, M. & GARCIA, J. (2007**).** *Comparison between genetic algorithms and the Baum-Welch algorithm in learning HMMs for human activity classification. Proceeding of EvoWorkshops'7*, 399–406.

[21] GOH, J., TANG, L., AL TURK, L. (2010). *Evolving the Structure of Hidden Markov Models for Micro aneurysms Detection*. *UK*

Workshop on Computational Intelligence (UKCI), 1–6.

[22] A. MAKHLOUF, L. LAZLI, B. BENSAKER, *Structure Evolution of Hidden Markov Models for Audiovisual Arabic Speech Recognition*, International Journal of Signal and Imaging Systems Engineering, IJSISE, 9(1), pp.55–66.

[23] S. YOUNG , G. EVERMANN, M. GALES, T. HAIN, D. KERSHAW, X. A. LIU, G. MOORE, J. ODELL, D. OLLASON, D. POVEY, V. VALTCHEV AND P. WOODLAND, *the HTK Book*, (for HTK Version 3.4). 2005.