# GMM Vector Quantization on the Modeling of DHMM for Arabic Isolated Word Recognition System

Snani Cherifa[1], Ramdani Messaoud[1], Zermi Narima[1], Bourouba Houcine[2]
[1]Laboratoire d'Automatique et Signaux d'Annaba (LASA)
Université Badji Mokhtar de Annaba
B.P 12, Annaba, 23000 Algeria
[2]Université de Guelma
BP 401, 24000, Guelma, Algérie

***ABSTRACT:*** *Vector quantization based on a codebook is a basic process to recognition the speech signal by discrete hidden markov model. This article identifies the fundamental importance of vector quantization codebooks in the performance of the system. For comparison, two different algorithms k-means and Gaussian mixture models (GMM) have been used to obtain two sets of speech feature codebook. We used in analysis phase Mel Frequency Cepstral Coefficients (MFCC) supplement by dynamic features to increase system performance, although experiments are carried out for the choice of the optimal parameters of the system. Good results are obtained using a GMM method.*

**Keywords:** Vector Quantization (VQ), Gaussian Mixture Models (GMM), K-means, Discrete Hidden Markov Model (DHMM), Speech Recognition System (SRS), Mel Frequency Cepstral Coefficients (MFCC)

**Received:** 29 May 2013, Revised 1 July 2013, Accepted 7 July 2013



## 1. Introduction

Speech is our most natural way of communicating. Effective integration of speech into man-machine communication can be accomplished by developing an Automatic Speech Recognition (ASR) system which allows a computer to identify the words that a person speaks into a microphone or telephone.

Speech recognition is a popular and active area of research [1] [2], used to translate words spoken by humans so as to make them computer recognizable. It usually involves extraction of patterns from digitized speech samples and representing them using an appropriate data model.

The most popular and successful approach to speech recognition is based on the Hidden Markov Model (HMM) [1] [2], which is a probabilistic process that models spoken utterances as the outputs of finite state machines. They have been applied in speech recognition because of their great adaptability and versatility in handling sequential signals.

Speech recognition systems contain tree main modules: feature extraction, training module and finally the classification or testing module. Feature extraction is the process that extracts a small amount of data from the speech signal that can later be used to represent each word. In the training phase, each speaker registered *N* samples of each word so that the system can build or train a reference model for that word. During the testing (operational) phase the input speech is matched with stored reference model (*s*) for each word and recognition decision is made. This paper presents a recognition system of Arabic isolated word

based on DHMM.

In order to use a discrete HMM, to reduce computation we must first quantize the data into a set of standard vectors, or a "*vector codebook*"; we used in this article two vector quantization methods k-means and GMM.

The outline of this paper is as follows. Section II describe the feature extraction based on MFCC, in Section III, we review the basics of a discrete Hidden Markov Model given the tree basics problems, Section IV describe isolated word recognition system and detailed the using of GMM in vector quantization steps. Section V experimental results obtained are detailed; finally section VI presents overall conclusions on the work.

## 2. The feature extraction (MFCC)

The main task of this stage is to convert speaker waveform into a more robust and compact representation that retains the information which is useful for discriminating speech and discard redundant information. The speech wave is usually analyzed based on spectral features.

### 2.1 Short-Term Features

Short term features are computed from short frames of about 20-30 milliseconds during which the speech signal is assumed to be stationary [3]. The development of this feature extraction approach is based on the fact that the speech signal continuously changes due to changes in the structure of the vocal tract. A sliding-window of 10-30 milliseconds is used to maintain the short term structure and to analyze the speech signal frame by frame.

### 2.2 Mel-Frequency Cepstral Coefficients (MFCC)

The most popular features used in speech recognition are mel-frequency cepstral coefficients (MFCCs) [2] [4] [5]. In this approach, a filter bank analysis is used such that the short-time spectrum of a speech signal is represented as a set of filter bank outputs. The filter bank is a series of band pass frequency filters which are defined by their frequency location (left frequency, central frequency, right frequency) and their shape. Spacing the filter bank according to a linear scale allow linear frequency cepstral coefficients (LFCC) to be extracted. However, a more common approach is to use a mel-scale. This scale transforms the linear frequency to the frequency scale of the human ear which has logarithmic nature. Figure1 shows the mel-scale as a function of linear frequency. As shown, this scale is nearly linear up to 1 KHz and is logarithmic for the higher frequencies. The central frequency of the filters spaced according to the mel-scale is given by [6]:

$$f_{mel} = 1000\, log_n(1 + f_{HZ} / 1000) / log_n 2 \qquad (1)$$

where, $f_{mel}$ is the mel-frequency and $f_{HZ}$ is the linear frequency.

Figure1 illustrates the extraction process required to obtain MFCC of a speech.

As shown in the figure 2, the speech signal is first pre-emphasized by applying a pre-emphasis filter. Pre-emphasis flatten the speech spectrum so as to reduce the dynamic range using a first order filter [2]

$$P(z) = 1 - 0,95\, Z^{-1} \qquad (2)$$

A window is then applied which allows the speech to be analyzed frame-by-frame. In speech recognition systems it is common to set the window length between 10-30 milliseconds [6]. This window is first applied at the beginning of the signal and is moved further until the end of the signal is reached. Often the overlap between consecutive windows is set 10 milliseconds [2]. Framing of the speech signal can be accomplished by multiplying the signal by a function that is constant inside a given interval and zero elsewhere. This is called rectangular window. Using this window introduces discontinuities at the frame edge which in turn leads to spectral leakage. A hamming window rather than rectangular window is the most used in speech recognition technology to taper the signal on the sides and thus reduce signal leakage because a high resolution is not required, considering that the next block in the all the closest frequency lines. Hamming window, whose impulse response is a raised cosine impulse has the form (3):

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1 \qquad (3)$$

Following the application of the window function, short-term spectral features are obtained by computing the discrete Fourier transform (DFT) of the speech contained within each frame, the discrete Fourier transform (DFT) is normally computed via the

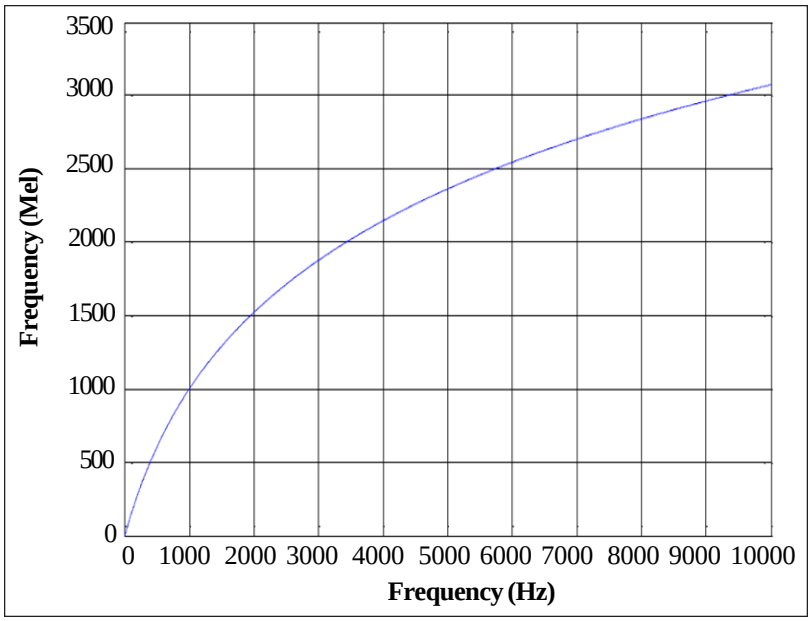


Figure 1. Mel-Frequency Scale

fast Fourier transform (FFT) algorithm, which is a widely used technique for evaluating the frequency spectrum of speech. FFT converts each frame of $N$ samples from the time domain into the frequency domain. The FFT is a fast algorithm, which exploits the inherent redundancy in the DFT and reduces the number of calculations.

Triangular filter bank spaced according to the mel scale is then applied to the speech spectrum. Such filters compute the average spectrum around each center frequency with increasing bandwidths, as displayed in Figure 3.

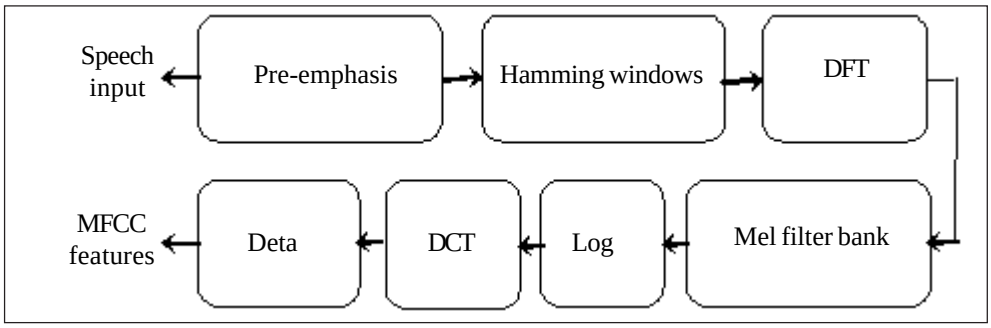


Figure 2. A block diagram of the structure of an MFCC extraction processor

After taking the logarithm of the filter bank output, cepstral coefficients are derived by applying discrete cosine transform (DCT) to the spectral vectors.

$$c_n = \sum_{m=1}^{M} [\log Y(m)] \cos\left(\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right) \qquad (4)$$

Where $n = 1, 2, \ldots, L$ and $Y(m)$, $m = 1, \ldots, M$ are triangular filter outputs and $L$ is the number of cepstral coefficients that we want to calculate ($L \leq M$).

**2.3 Delta and Delta Delta Coefficients**

MFCC coefficients are static features as they only give information about the speech signal over a fixed period of time. In order to capture information about how this vectors change over time, dynamic coefficients called delta and delta-delta coefficients are appended. These dynamic features track the temporal variability in the feature vectors and are found to improve the

recognition accuracy. The delta coefficients are computed via linear regression over a window of consecutive frames with window length of 3-7 seconds [7].

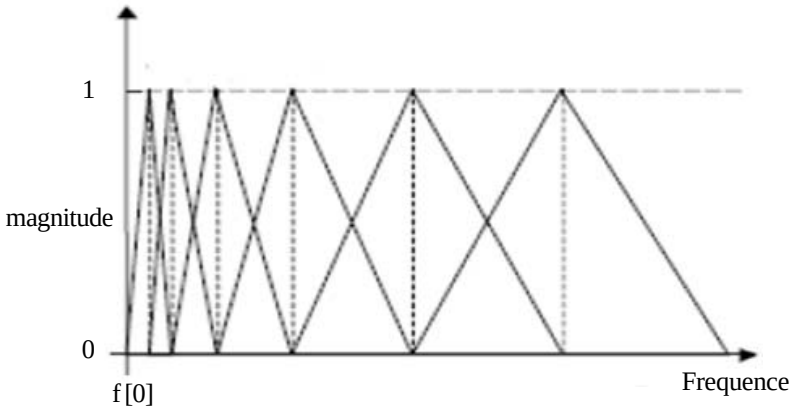


Figure 1. Triangular filters used to compute Mel-cepstrum

$$\Delta C_t = \frac{\sum_{k=1}^{K} (C_{t+k} - C_{t-k})}{2\sum_{k=1}^{K} k^2} \qquad (5)$$

Where $c$ and $\Delta c$ correspond to the static and delta (dynamic) coefficients respectively, $K$ is the number of surrounding frames and $c_t$ is the feature vector for which the delta coefficients are being computed for. Double delta (delta-delta) coefficients are computed in a similar way over the delta coefficients.

### 3. Presentation OF DISCRET hidden makov model (HMM)

HMMs are doubly stochastic models capable of statistical learning and classification. An HMM model is a finite state machine that changes state at every time unit. At each discrete time instant $t$, transition occurs from state $i$ to $j$, and the observation vector $o_t$ is emitted with the probability density $b_j(o_t)$. The output probability distributions can be either discrete or continuous. Moreover the transition from state $i$ to $j$ is also random and it occurs with the probability $a_{ij}$.

A good introduction to hidden Markov model can be found in [8].

#### 3.1 Elements of an DHMM

The complete description of the model can be provided using the following quantities [2] [8]:

- $N$, the number of states in the model. We denote the individual states as $S = \{S_1, S_2, ..., S_N\}$ and the state at time $t$ as $q_t$.
- $M$, the number of distinct observation symbols per sate. We denote the individual symbols as $V = \{v_1, v_2,..., v_M\}$
- The state transition probability distribution $A = \{ a_{ij} \}$ where $a_{ij} = p\,(q_{t+1} = S_j \setminus q_t = S_j) \quad 1 \le i, j \le N$
- The observation symbol probability distribution in state $j$, $B = \{b_j(k)\}$, where $b_j(k) = p\,(v_k \text{ at } t \setminus q_t = S_j) \quad 1 \le j \le N, 1 \le k \le M.$
- The initial state distribution $\pi = \{ \pi_i \}$ Where $\pi_i = P\,[q_1 = S_i]\ 1 \le i \le N.$

Given appropriate values of $N$, $M$, $A$, $B$, and $\pi$, the HMM can be used as a generator to give an observation sequence $O = \{O_1, O_2, O_T\}$, where each observation $O_t$ is one of the symbols from $V$, and $T$ s the number of observations in the sequence.

We use the compact notation $\lambda = (A, B, \pi)$ to indicate the complete parameter set of the model.

#### 3.2 Three basic problems for HMMs

##### 3.2.1 Evaluation

Given the observation sequence $O = \{O1, O2, ..., OT\}$ and the model $\lambda$.

How do we compute $p\ (O \mid \lambda)$ = the probability of sequence O being generated by the model. To know which model better represents O ?⇒ recognition

**3.2.2 Segmentation**

Given the observation sequence $O = \{O1, O2, ..., OT\}$ and model ë

How do we choose a state sequence $Q = \{q1, q2, ..., qT\}$ which is optimum in some sense?

**Training or estimation:** Given the observation sequence $O = \{O1, O2, ..., OT\}$.

How do we adjust the model parameters $\lambda$ to maximize $p\ (O \mid \lambda)$?

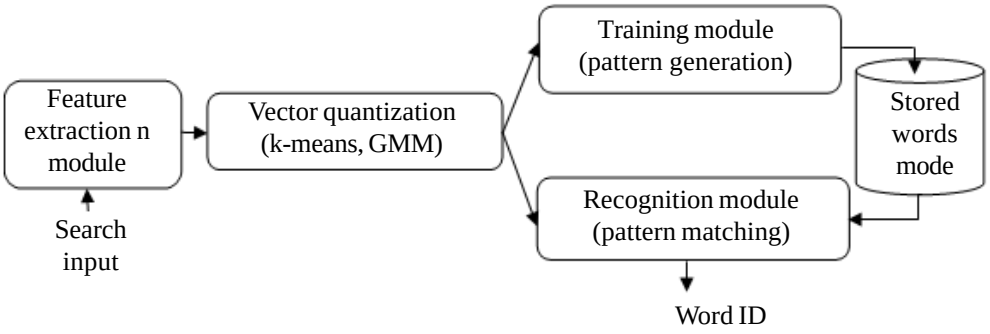


Figure 4. Block diagrams of an isolated word DHMM recognize

**Objective:** optimize $\lambda$ parameters to better describe the sequence

**4. Recognition of Arabic isolated words based on discrete HMMs**

The algorithm of recognition of the isolated words based on DHMMs is summarized in three principal stages as shown in figure4: the first one is the feature extraction (MFCC) phase which the speech signal is transformed to a set of spectral vectors, Details of this transformation was given in section II. While the second one is referred to the enrollment sessions or training phase and the last one is referred to as the operation sessions or testing phase, the last two stages are detailed in this section:

**4.1 Modeling process**

A discrete HMM isolated word recognition system can be described as a two-step modeling process.

**4.1.1 The first step, vector quantization (VQ)**

In order to use a discrete HMM, to reduce computation we must first quantize the data into a set of standard vectors, or a "*vector codebook*". VQ [2] [10] [11] is used to classify the speech signal space into *N* region (cluster), where *N* is the codebook size or number of models generated in this step. Each region is represented by a typical vector, usually the centroid vector for that region. The codebook is then composed of these typical vectors, we used for this step two classification methods k-means and GMM.

**4.1.1.1 Algorithme k-means**

K-means algorithm [12], [13] [14] was originally designed for vector quantization codebook generation. It is an unsupervised clustering algorithm, which represents each cluster with its mean. Assuming a set of vectors $X = \{\bar{x}_1, \bar{x}_2, \bar{x}_3, ..., \bar{x}_T\}$ which is divided into *M* clusters represented by their mean vectors $\{\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3, ..., \bar{\mu}_M\}$, the objective of the K-means algorithm is to minimize the total distortion given by equation (6):

$$Total\ distortion = \sum_{t=1}^{T} \sum_{L=1}^{M} \| \bar{x}_L - \bar{\mu}_L \| \qquad (6)$$

*K*-means follows an iterative approach to meet the objective. In each successive iteration, it redistributes the vectors in order to minimize the distortion. Although originally meant for codebook generation, it can be adapted to train GMM. The procedure consists of four steps as shown in figure 6:

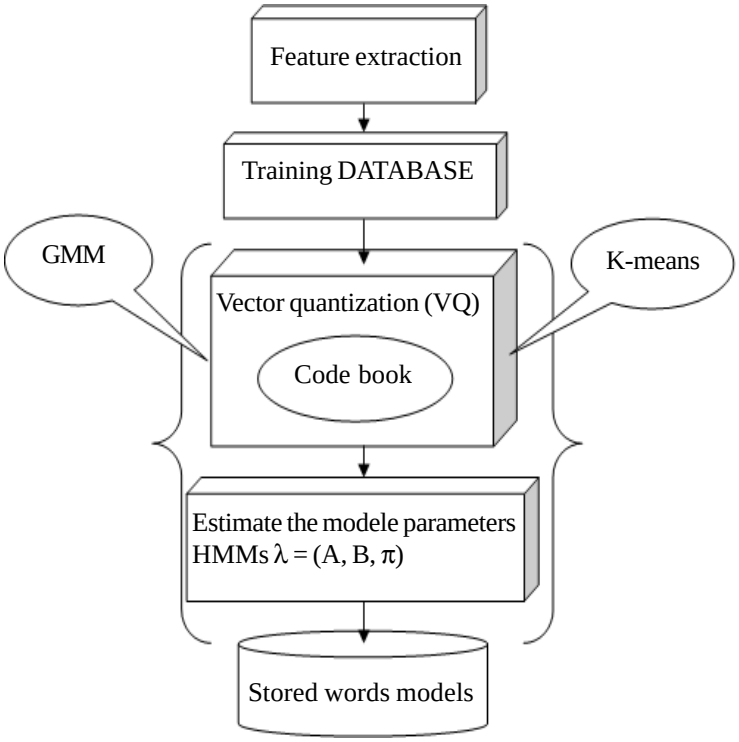


Figure 5. A block diagram of the training processor

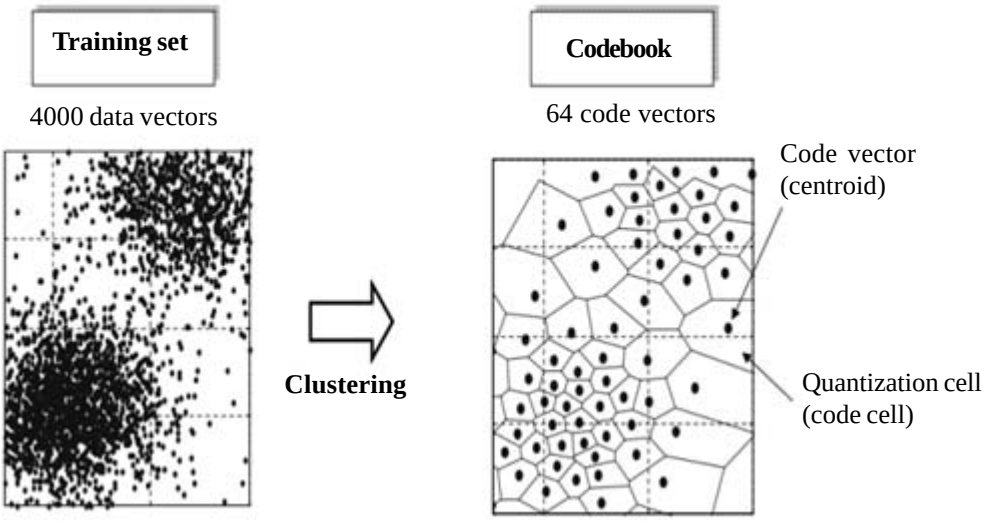


Figure 6. Codebook construction of VQ using K-means algorithm [1]

• To initialize, *M* random vector from the training set are selected as the means of *M* clusters.

• Each vector $\overline{x}_t$, $1 \le t \le T$ is assigned to cluster *j*, if,

$$\| \overline{x}_L - \overline{\mu}_L \| < \| \overline{x}_L - \overline{\mu}_k \| \; \forall \; k \ne j, 1 \le j, k \le M \qquad (7)$$

• The new 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.

The first and second steps are repeated till the clusters are stable, i.e., the distortion is minimized. When the clusters are stable, the weights and covariance matrix can be found as described in step *d*. It is to be noted that in each iteration, *K*-means estimates the means of all the *M* clusters.

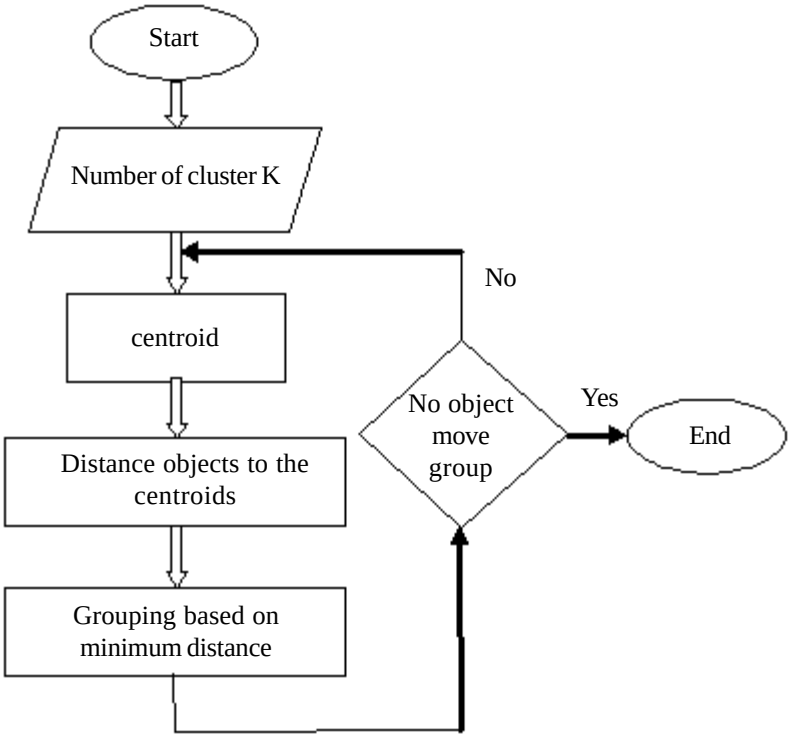


Figure 7. K-means algorithm

**4.1.1.2 Gaussian mixture models Vector quantization (GMMs)**

In this paper, we use Gaussian mixture models for the speech data. Considering each model in the mixture as a cluster of the data space, we in fact use the EM algorithm to classify the signal space into clusters of which each cluster is represented by a model in the mixture. In other words, the EM algorithm performs a vector quantization and generates a codebook in which each code word is a Gaussian model composed of a mean vector, a covariance matrix, and a mixture weight. It is obvious in such a vector quantizer that the EM classifier decodes the speech feature vectors in a stochastic sense, and carries this stochastic information into the HMM.

**4.1.1.2.1 The guassian Model Description) [15]**

In the GMM model, the features distributions of the speech signal are modeled for each cluster as follows.

$$p(x \mid \lambda) = \sum_{i=1}^{M} p_i b_i(x) \quad (8)$$

where

$$\sum_{i=1}^{M} p_i = 1 \quad (9)$$

$x$ is a random vector of D-dimension, $p(x/\lambda)$ is the region model; $p_i$ is the $i^{th}$ mixture weights; $b_i(x)$ is the ith pdf component that is formed by the $i^{th}$ mean $\mu_i$ and $i^{th}$ covariance matrix, where $i = 1, 2, 3, \ldots, M$, and $M$ is the number of GMM components, each density component is a D-variants Gaussian distribution given equation (11) Description of the GMM system herein uses the same notation as in [15].

A statistical model for each cluster $S$ in the set is developed and denoted by $\lambda$. For instance, cluster $s$ in the set of size $S$ can be

written as follows:

$$\lambda_s = \{p_i, \vec{\mu}_i, \Sigma_i\}, \quad i = (1,.....M) \tag{10}$$

Where $\vec{x}$ D-dimensional random vector

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{ \frac{1}{2} (\vec{x}_i \ \vec{\mu}_i)' \Sigma_i \ (\vec{x}_i \ \vec{\mu}_i) \right\} \tag{11}$$

**4.1.1.2.2 ML Parameter Estimation**

Given training speech (transformed to spectral vectors) from a speaker's voice, the goal of cluster model training is to estimate the parameters of the GMM $\lambda$, which in some sense best matches the distribution of the training feature vectors.

The most popular method for training GMMs is a maximum likelihood (ML) estimation [15]. The aim of ML estimation is to find the model parameters, which maximize the likelihood of the GMM given the training data. For a sequence of *T* training vectors $X = (x_1,..., x_T)$ the GMM likelihood can be written as:

$$p(X|\lambda) = \prod_{i=1}^{T} p(x_i|\lambda) \tag{12}$$

Maximization of the quantity in (10) is accomplished through running the expectation-maximization (EM) algorithm.

In which EM steps can be given as:

• Beginning with an initial model $\lambda$

• Estimate a new model $\bar{\lambda}$ such that
$p(X|\bar{\lambda}) \geq p(X|\lambda)$

• Repeated 2. until convergence is reached

Following formulas are used on each EM iteration.

Mixture Weights

$$\bar{p}_i = \frac{1}{T} \sum_{t=1}^{T} p(i|\vec{x}_t, \lambda) \tag{13}$$

Means

$$\bar{\vec{\mu}}_i = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda) \ \vec{x}_t}{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)} \tag{14}$$

Variances

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda)} \ \bar{\mu}_i^2 \tag{15}$$

The *posteriori* probability for acoustic class is given by

$$p(i|\vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} p_k b_k(\vec{x}_t)} \tag{16}$$

**4.1.1.2.3 Vector quantization by GMM**

A group of clusters $S = \{1, 2,\ldots, s\}$ is represented by GMM's $\lambda_1, \lambda_2, \ldots, \lambda_s$

The objective is to find the cluster model, which has the maximum *a posteriori* probability for a given observation sequence for each word.

$$\hat{s} = \arg \max_{1 \leq k \leq S} Pr(\lambda_k|X) = \arg \max_{1 \leq k \leq S} \frac{p(X|\lambda_k) \ Pr(\lambda_k)}{p(X)} \tag{17}$$

Where the second equation is due to Bayes's rule. Assuming equally likely clustres ($P(k)$ $1/S$) $\lambda$ = and noting that $p(X)$ is the same for all cluster models, the classification becomes:

$$\hat{s} = \arg \max_{1 \le k \le S} Pr(X \mid \lambda_k) \tag{18}$$

Finally with logarithms, the speaker identification system gives:

$$\hat{s} = \arg \max_{1 \le k \le S} \sum_{t=1}^{T} \log p(\vec{x}_t \mid \lambda_k) \tag{19}$$

Which

$$p(\vec{x}_t \mid \lambda_k) = \sum_{t=1}^{T} p_k b_k(\vec{x}_t) \tag{20}$$

**4.1.2 The second step, HMM building**

Is used to produce a set of reference models that represent the possible sequences of the quantized observation vectors from the codebook generated by the first step, where for each word $V$ in the vocabulary, we must build an HMM noted $\lambda v$, we must estimate the model parameters $\lambda = (A, B, \pi)$ that optimize the likelihood of the training set observation vectors for the 5th word. In this stage we applied calls iterative algorithms of estimation of Baum-Welch.

**4.2 Recognition**

For each unknown word which is to be recognized, the processing of figure.8 must be carried out, namely measurement of observation sequence $O = \{O_1, O_2, ..., O_T\}$, via a feature analysis of the speech corresponding to the word; followed by calculation of model likelihoods for all possible models, $p(O \mid \lambda v)$, $1 \le v \le V$, followed by selection of the word whose model likelihood is highest.

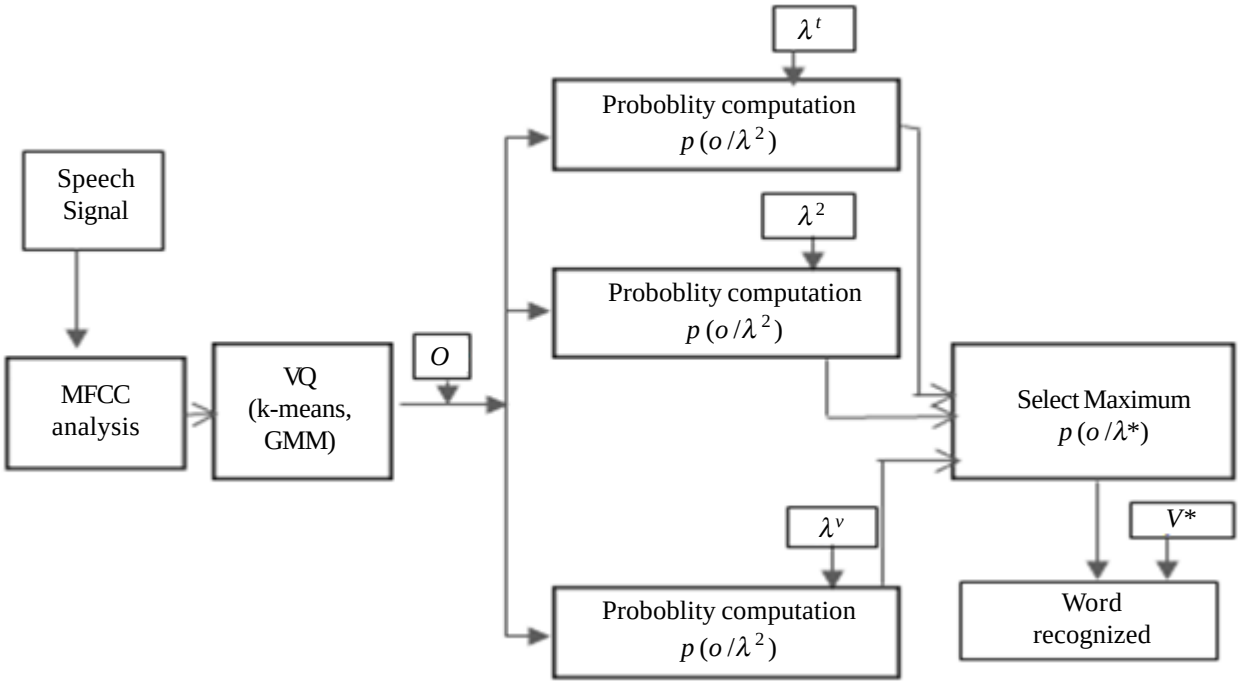


Figure 7. A block diagram of the recognition processor

**5. Results and discussion**

Different stages of a speech recognition system were designed and implemented using matlab 6.5 we used the left-right model of R.Bakis.

In the flowing we present the results of an experimental study aimed at finding the effect of the quantization methods (k-means and GMM) on the recognition performance. Using LASA database

• **Training DataBase:** For the training phase we built a base BA containing 10 speakers each speaker recorded a 7 utterances of each word. BA is used to produce a set of reference models and code book generation, we choose number of code book equal $M$=128.

• **Testing DataBase:** For the testing phase we built a base BT containing 10 speakers each speaker recorded 3 utterances for each words.

The speech signal for each speaker is first sampled at 11.025 kHz sampling rate and pre emphasized using a fixed $1^{rst}$ order digital system with transfer function: $H(z) = 1 - 0.95Z^{-1}$ . The signal is next blocked into 256 sample sections (< 30ms for each frame, the overlap ration between successive frames is 0.5, after each frame is multiplied by Hamming window $w(n)$. Finally for each frame, we are computed 4 vectors of MFCC coefficients:

• V1: 12 MFCC coefficients $\Rightarrow$12 Coeffs for each frame.

• V2: 12 MFCC coeffs +energy coeff $\Rightarrow$13 Coeffs for each frame.

• V3: 12 MFCC coeffs + energy coeff + Delta (12 MFCC coeffs + energy coeff) $\Rightarrow$ 26 Coeff for each frame.

• V4: 12 MFCC coefficients + energy coefficient + Delta (12 MFCC coeffs + energy coeff) + DeltaDlta (12 MFCC coeffs + energy coeff) $\Rightarrow$ 39 Coeffs for each frame.

| **word** | **k-means** | **GMM** |
|---|---|---|
| 0 | 93 | 94 |
| 1 | 96 | 97 |
| 2 | 93 | 98 |
| 3 | 100 | 98 |
| 4 | 93 | 92 |
| 5 | 100 | 99 |
| 6 | 91 | 99 |
| 7 | 92 | 94 |
| 8 | 98 | 94 |
| 9 | 97 | 99 |
| average rate % | 94.30 | 96.4 |

Table 1. Choice of the quantization method

| average rate % using | **k-means** | **GMM** |
|---|---|---|
| V1 | 82.62 | 84.23 |
| V2 | 85.09 | 88.10 |
| V3 | 86.8 | 89.2 |
| V4 | 89.2 | 91.6 |

Table 2. Conclusion And Future Work

In the implementation of DHMM on speech recognition platform, the number of hidden states can be arbitrarily set and the number of observation is set to be the number of the cluster in the codebook which is used to quantize the speech signal to be identified. In this application, we use 5 hidden states, 128 observations (the number of clusters in codebook) for DHMM. Consequently, the dimension of the codebook in this experiment is $128 \times 10$. Moreover, the dimension of the matrices $A$, $B$, and $\pi$, in the DHMM model described in section.2 are $5 \times 5$, $5 \times 128$, and $5 \times 1$, respectively.

The words which will be recognized are 0 – 9. And hence there are 10 DHMM models after the training phase.

Result in table. 1 show a better average rate of recognition obtained by the use of GMM vector quantization method.

Table. 2 results reveal that the speech recognition rate can be improved by using dynamic features in analysis phase.

In conclusion we can say as the quality of the quantization method determines the performances of the system of recognitions.

## 6. Conclusion And Future Work

In this paper we present several techniques used in the design of an arabic isolated words recognition system based on discrete hidden markov model. in the training phase we study two methods of quantization: gaussian mixture model (gmm) and k-means, The best rate of recognition (96.4%) is obtained by the use of the GMM method. we also study the performance of the system using energy and dynamic coefficients as supplementary coefficients of mel frequency cepstral coefficients (MFCC) in analysis phase, although experiments are carried out for the choice of the optimal parameters of the system. good results are obtained using a GMM method. Feature work focus to use hierarchical gaussian mixture as clustring algorihm for DHMM.

## References


[1] Tao, J., Xin, L., Yin, P. (2009). Realistic visual speech synthesis based on hybrid concatenation method, *IEEE Trans. on Audio, Speech, and Language Processing*,17 (3) 469-477.

[2] Huang, X., Acero, A., Wuenon, H. (2005). Spoken Language Processing A Guide to Theory, Algorithm and System Developmen. *Pearson*.

[3] Deller, J. R., Hansen, J. H. L., Proakis, J. G. (2000). *Discrete-Time Processing of Speech Signals*, Piscataway (N.J.), *IEEE Press*.

[4] O'Shaughnessy, D. (2003). Interacting with computers by voice: Automatic speech recognition and synthesis, *In*: Proc. IEEE, 91 (9), Nov.

[5] Davis, Paul Mermelstein, Steven, B. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech, and Signal Processing.* 28, p. 357-366.

[6] Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacretaz, D., Reynolds, D. A. (2004). A Tutorial on Text-Independent Speaker Verification. *EURASIP Journal on Applied Signal Process.* p. 430-451

[7] Campbell, W. M., Campbell, J. P., Reynolds, D. A., Jones, D. A., Leek, T. R. (2004). High-level speaker verification with support vector machines. *In*: Proceedings of the IEEE International Conference on Acoustics, *Speech and Signal Processing*, (*ICASSP*'04). 1, p. 73-76.

[8] Rabiner, L. R. A Tutorial on Hidden Markov models and Selected Application in Speech Recognition, *In*: Proc. IEEE, 7 (2) 257-286, Feb 89.

[9] Boite, R., Bourlad, H., Dutoit, T., Hancq, J., Leich, H. *Traitement de la parole,* presses polytechniques et universitaires romandes, France, décembre.

[10] Furui, S. (1991). Vector-Quantization-Based Speech Recognition and Speaker Recognition Techniques. *IEEE Signals, Systems and Computers*, 2, p. 954-958.

[11] Manjot Kaur Gill, Reetkamal Kaur, Jagdev Kaur. (2010). Vector Quantization based Speaker Identification. *International Journal of Computer Applications.* 4 (2).

[12] Linde Y., Buzo A., Gray R. M. (1980). An Alogorithm for Vector quantizer, *IEEE Transactions on Communication*, 28 (1).

[13] Singh, G., Panda, A., Bhattacharyya, S., Srikanthan, T. (2003). Vector Quantization Techniques for GMM Based Speaker Verification, *ICASSP*.

[14] San, O. M., Huynh, V. N., Nakamori, Y. (2004). An Alternative Extension Of The K-Means Algorithm for Clustering Categorical Data, *Int. J. Appl. Math. Comput. Sci.*, 14 (2) 241– 247.

[15] Reynolds, Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3 (1) 72–83.