

Performance Tests on Several Parametric Representations for An Arabic Phoneme Recognition System using HMMs

Iman A.Maaly and Mohamed A. H. Abbas
Department of Electrical Engineering
University of Khartoum
SUDAN

Abdel R. Elobeid
Physics Department, Faculty of Science
University of Qatar
QATAR

Abstract

An Arabic phoneme recognition system using Hidden Markov Models (HMM) is introduced. This system is an important step towards the realization of a continuous speech recognition system with a large size of Arabic vocabulary. A discrete HMM is implemented for modeling each of the Arabic phonemes. Training and recognition are both based on Viterbi methods. For deciding on the best features that can represent Arabic speech signals, performance tests were implemented on a number of parametric representations such as prediction coefficients, area function, cepstral coefficients, etc. Results showed the superiority of the cepstral coefficients, with a recognition performance of 74% over the other representations. Results also showed that supplementing the cepstral coefficients with delta power and delta-delta power improves the performance to 81%.

Key words: Speech coding, speech recognition, and HMM.

1. Introduction

The objective of achieving a robust, intelligent, fluent spoken Arabic recognition machine still remains a distant goal. This is perhaps mainly due to insufficient amount of studies conducted in this field in relation to the Arabic language and Arabic speech.

Arabic speech has its distinctive features, e.g., it has a large number of pharyngeal sounds (e.g., /x/, /ʔ/) and a large number of emphatic ones (e.g., /x/). It also has the two palatal sounds /t/ and /d/ that are rarely used in other languages. Also, nowadays the existence of many Arabic dialects raise a problem of Arabic phonetic study. Modern standard Arabic is the type of

Arabic most often used in education and used by the announcers in broadcasting corporations in the Arab world. But this is still affected by the dialect of each country. For this reason performance tests are implemented to find the best parameters that can characterize this language.

Although there is some disagreement among linguists [1]- [4], modern standard Arabic is generally considered to have around 34 phonemes. These 34 phonemes comprises 28 consonants, 3 short vowels and 3 long vowels. The 3 short vowels have the same characteristics of the long ones but with shorter duration.

corresponding to 20 msec of signal. Total number of frames is denoted N_f . The combination of a 20 ms frame duration and (5 ms) overlap is used in this system.

2.3 Signal Analysis

The LPC analysis technique is implemented in this recognition system to obtain a set of parameters that represent speech signals [5], [6], [9], [10], (see Fig. (1)). The covariance lattice method of LPC is used because it guarantees the stability of the LP filter with moderate computational effort and without windowing [9]. Linear prediction methods model the signal spectrum by an all-pole filter with a transfer function given by

$$H(Z) = \frac{G}{A(Z)} = \frac{G}{1 + \sum_{k=1}^P a_k Z^{-k}} \quad (2)$$

Where G is a gain factor,
 $\{a_k\}$ are the predictor coefficients,
 P the prediction number,
 $A(Z)$ is the inverse filter that can be implemented as a Lattice filter.

According to the structure of the lattice network, given a signal $s(n)$, the covariance matrix at each stage m is computed using the following equation

$$\phi_m(k, i) = \sum_{n=m}^{N_f-1} s(n-k)s(n-i), \quad (3)$$

$$0 \leq k, i \leq m$$

This covariance matrix, $\phi_m(k, i)$, is used in computing the reflection coefficients k_m by minimizing the error criterion as follows

$$F_m(n) = \sum_{k=0}^m \sum_{i=0}^m a_k^{(m)} a_i^{(m)} \phi(k, i)$$

$$B_m(n-1) = \sum_{k=0}^m \sum_{i=0}^m a_k^{(m)} a_i^{(m)} \phi(m+1-k, m+1-i)$$

$$C_m(n) = \sum_{k=0}^m \sum_{i=0}^m a_k^{(m)} a_i^{(m)} \phi(k, m+1-i)$$

$$k_m = \frac{-2C_m(n)}{F_m(n) + B_m(n-1)} \quad (4)$$

Finally, the reflection coefficients k_m in the lattice are uniquely related to the predictor coefficients. We used the following recursive relation for obtaining a_m from k_m .

$$a_m^{(m)} = k_m$$

$$a_j^{(m)} = a_j^{(m-1)} + k_m a_{m-j}^{(m-1)} \quad (5)$$

and $1 \leq j \leq m-1$

Where $a_j^{(m)}$ are the prediction coefficients at stage m . The equations in (5) are computed recursively for $m = 1, 2, \dots, P$, and the final solution is given by

$$a_j = a_j^{(P)}, \quad 1 \leq j \leq P \quad (6)$$

Choice of the predictor number P depends primarily on the sampling rate f_s and is essentially independent of the LPC method used [5]. It is found that the value of P which gives the best results is $P = 13$ [5], [8]. So, for each

frame (l) a set of 13 LPC coefficients denoted¹ $\{a_l(i)\}$ plus a set of 13 reflection coefficients denoted $\{k_l(i)\}$ are computed (where $1 \leq i \leq P$ and $1 \leq l \leq N_f$).

2.4 Cepstral Coefficients

We obtain the cepstral coefficients from the predictor coefficients as follows:

$$c_l(1) = -a_l(1) \quad (7)$$

$$c_l(i) = -a_l(i) - \sum_{j=1}^{i-1} \left(1 - \frac{j}{i}\right) a_l(i) c_l(i-j) \quad (8)$$

Where $\{c_l(i)\}$ is the LP-derived cepstral vector at frame (l), $2 \leq i \leq N_c$.

The number of cepstral coefficients N_c is usually such that

$$0.75 P \leq N_c \leq 1.25 P$$

where P is the predictor number [7]. We have chosen in this system $N_c = P = 13$.

2.5 The Liftering Process

The liftering process is then implemented on the cepstral coefficients for the purpose of weighting them and to enhance those portions of the cepstrum representing the vocal tract information. This process is defined as follows [7]:

¹ In next sections, the subscript will denote frame number (l) and index of prediction parameter will be included between brackets.

$$\hat{c}_l(m) = c_l(m) W_c(m) \quad (9)$$

$$W_c(m) = 1 + \frac{N_c}{2} \sin \frac{\pi m}{N_c}, \quad 1 \leq m \leq N_c \quad (10)$$

Where $W_c(m)$ is the weighting or window function.

2.6 Parameter Transforms

Parameter transforms are accomplished through a differentiation process. We used the following first order differentiator [7], [11]

$$\Delta \hat{c}_l(m) = \sum_{n=-N_d}^{N_d} n \hat{c}_{l-n}(m) \cdot D \quad (11)$$

Where D is a gain term chosen to make the variances of $\hat{c}_l(m)$ and $\Delta \hat{c}_l(m)$ equal (a value for D of 0.375 is used [11]), and $N_d = 1$ (first-order differentiator). A 3-frame window is used in the computations of the derivative. The signal output from this differentiation process is denoted a delta parameter. The second-order derivative ($\Delta \Delta \hat{c}_l(m)$) is obtained by reapplying (11) to the output of the first order differentiator.

2.7 Area Function

The reflection coefficients that are obtained directly from the LP analysis technique, are used in a direct and simple formula to obtain the area function as follows [12],[13]

$$A_l(P+1) = 1 \quad (12)$$

$$A_l(m) = \frac{1+k_l(m)}{1-k_l(m)} A_l(m+1) \quad (13)$$

Where $k_l(m)$ is the m^{th} reflection coefficient of frame l , and $1 \leq m \leq P$.

2.8 Log Area Ratio Parameters

An important set of equivalent parameters which is derived from the reflection coefficients is the log area ratio parameters which are calculated as follows [5]

$$\begin{aligned} g_l(m) &= \log\left(\frac{1+k_l(m)}{1-k_l(m)}\right) \\ &= \log\left(\frac{A_l(m)}{A_l(m+1)}\right) \end{aligned} \quad (14)$$

where $1 \leq m \leq P$

2.9 Formant Frequencies

The formant frequencies are obtained by measuring the frequency of the peak in the spectral envelope [14], [5]. As a first step, the spectral envelope, which is the power spectrum of the impulse response of the linear prediction filter, is calculated as follows,

$$G(f) = \frac{1}{\left|1 - \sum_{k=1}^P a_k e^{-2\pi j k f / f_s}\right|^2} \quad (15)$$

Where $G(f)$ is the spectral envelope at frequency f , $\{a_k\}$ are the predictor coefficients, and f_s is the sampling frequency.

Formant frequencies are then extracted by detecting the local maxima by

magnitude comparison of all the spectral samples. The Formants in the spectrum are usually denoted F_1, F_2, F_3, \dots beginning with the lowest frequency.

2.10 Power

Power is also computed on a frame-by-frame basis, see Fig. (2), as follows :

$$P_n = \frac{1}{N_w} \sum_{m=0}^{N_w-1} \left[W(m) s\left(n - \frac{N_w}{2} + m\right) \right]^2 \quad (16)$$

Where N_w is the window duration in samples ($N_w=300$), $W(m)$ is the Hamming window, n denotes the sample index of the center of the window which corresponds to frame l , and $s(n)$ denotes the signal. The Hamming window is defined as

$$W(m) = \frac{0.54 - 0.46 \cos(2\pi m / (N_w - 1))}{\beta_w} \quad (17)$$

for $0 \leq m \leq N_w$, and $W(m) = 0$ elsewhere.

β_w is a normalization constant in the range (0,1) defined so that the root mean square (rms) of the window is unity. i.e.,

$$\beta_w = \sqrt{\frac{1}{N_w} \sum_{m=0}^{N_w-1} W^2(m)} \quad (18)$$

By implementing (16) a single value of the power is obtained for each frame of the signal [7], [15].

2.11 Power Differentiation

For obtaining the delta power parameters, the same first order differentiator as in (11) is used as follows

$$\Delta P_l = \sum_{n=-N_d}^{N_d} n P_{l-n} \cdot D \quad (19)$$

We obtain the second order derivative ($\Delta\Delta P_l$) of power by reapplying (19) .

3. Statistical Modeling and Vector Quantization

In this section we consider joint quantization of a block of signal parameters. This type of quantization is called block or Vector Quantization (VQ) [6], [7], [16], [17] which is implemented as a compression technique to reduce the computational complexity. The output of this stage is a set of observation vectors representing the signal. These observation vectors are used in the HMM recognition system. The first step in this process is the initial guess of the code book. Initial guess by splitting is implemented on this system [16] .

We have an N_f -dimensional random vector

$$Y = [Y_1 Y_2 \dots Y_{N_f}] \quad (20)$$

Where N_f is the number of time frames. Y_l are real random variables which represent the l^{th} frame of speech. These random variables correspond to the parameter set

extracted from speech signals as follows:

$$Y_l = [y_l(1) y_l(2) y_l(3) \dots y_l(N_v)]^T \quad (21)$$

Where N_v is the dimension of the parameter set.

Formally the quantizer maps the random vector Y to another random vector X of dimension N_f

$$X = [X_1 X_2 \dots X_{N_f}] \quad (22)$$

This mapping can be expressed as

$$X = Q(Y) \quad (23)$$

The vector X has a special distribution in that it may only take one value of N_{vq} values, its *pdf* will consist of N_{vq} impulses over the N_f -dimensional hyper-plane. Let us designate these values by $v_1 v_2 \dots v_{N_{vq}}$.

Quantizing the vector Y into the vector X introduces a quantization error or a distortion $d(Y,X)$. The distortion measure $d(Y,X)$ is a distance measure. The Euclidean distance measure is used for measuring this distortion as follows

$$d(Y,X) = \sqrt{\sum_{k=1}^{N_f} |y_k - x_k|^2} = \sqrt{[Y - X]^T [Y - X]} \quad (24)$$

The vector quantization algorithm is implemented using the LBG method as follows [6], [7],[16],

1. Initialization:

