Text-Independent Speaker Verification Based on Information Theoretic Learning

SHEERAZ MEMON*, TARIQ JAMEEL SAIFULLAH KHANZADA*, AND SANIA BHATTI**

RECEIVED ON 01.02.2011 ACCEPTED ON 07.06.2011

ABSTRACT

In this paper VQ (Vector Quantization) based on information theoretic learning is investigated for the task of text-independent speaker verification. A novel VQ method based on the IT (Information Theoretic) principles is used for the task of speaker verification and compared with two classical VQ approaches: the K-means algorithm and the LBG (Linde Buzo Gray) algorithm. The paper provides a theoretical background of the vector quantization techniques, which is followed by experimental results illustrating their performance. The results demonstrated that the ITVQ (Information Theoretic Vector Quantization) provided the best performance in terms of classification rates, EER (Equal Error Rates) and the MSE (Mean Squared Error) compare to K-means and the LBG algorithms. The outstanding performance of the ITVQ algorithm can be attributed to the fact that the IT criteria used by this algorithm provide superior matching between distribution of the original data vectors and the codewords.

Key Words: **Speaker Verification, Kmeans, LBG, Information Theoretic Learning, Vector Quantization.**

1. INTRODUCTION

The VQ method is a classical signal processing technique which models the probability density functions by the distributions of prototype vectors. A typical VQ algorithm divides a large set of vectors into clusters having number of points. Each cluster is represented by its central point. According to the Shannon's rate distortion theory [1], the central points for each cluster should be calculated as centers of gravity (or centroids); and the cluster members should be ideally selected such that, for each cluster member, the cluster centroid is the nearest centroid. The VQ techniques have been widely adapted as speaker modeling techniques in speaker recognition or verification tasks [2-3]. A VQ technique encompasses two fundamental tasks:

(1) An encoding process which involves a NN (Nearest Neighbor) search, assigning the closed codeword to a given vector.

(2) A codebook generation process which finds an optimal, small set of vectors (codebook) representing a given large set of vectors. The elements of codebook are called the codewords.

* Assistant Professor, Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro.
** Assistant Professor, Department of Software Engineering, Mehran University of Engineering & Technology, Jamshoro.

The best known and very efficient VQ codebook generation algorithm used in speaker verification task includes: the K-means algorithm [4], the LBG algorithm [5], and the KSOM (Kohonen's Self Organizing Map) [6]. In these algorithms the process of finding an optimal codebook is guided by minimization of the average distortion function (objective or cost function) representing an average total sum of distances between the original vectors and the codewords. It is also called the quantization error. Different types of distance measures for the quantization error have been proposed in literature [1]. The VQ codebook generation is a large scale global optimization problem, however the vast complexity of this problem means that in reality only sub-optimal solutions can be found. Codebook generation algorithms are distinguished on the bases of finding acceptable local minima of the objective function. An ideal codebook should contain a set of uncorrelated (linearly independent) centroid vectors. In reality there is always remaining a certain amount of correlation between centroids.

VQ may be used as a classification process in a number of ways [3]. The most often used approach is to generate a separate codebook for each speaker using speech recordings that belong to that speaker. During the testing phase the set XID of observed feature vectors from the unknown speaker are compared with codebooks representing the reference speakers. This process is graphically illustrated in Fig. 1 [7]. The quantization errors for the observed feature vectors of the unknown speaker are used as a measure of how close the observed feature vectors are to codewords representing each speaker. The speaker whose codebook is the closest to the observed feature vectors is then taken as the identified speaker. In speaker verification task, an arbitrary threshold is often applied to the quantization error to determine if the observed feature vectors are close enough to the codebook for the claimant speaker to accept the claim. In [8] we evaluated ITVQ for TIMIT speech corpus; in [9] we used ITVQ with GMM to improve the classification rates. This paper is further organized as follows: In Section 2 and 3 we describe the conventional K-means and LBG algorithms, Section 4 describes the ITVQ procedure to compute speaker models, it is followed by various evaluation tests conducted in Section 5, and in Section 6 we conclude our work.

2. K-MEANS CLUSTERING

The K-means [4,10] clusters data based on attributes or features into K groups where, K is a positive integer. The clustering is achieved by minimizing the squared Euclidean distance between vectors x_i and the corresponding cluster centroid vector θ_j. The centroid vector represents each cluster as a mean vector of the cluster. Let us assume that a set of T vectors $X=\{x_1,x_2,x_3,\ldots,x_T\}$ is to be divided into K clusters represented by their mean vectors $\theta=\{\theta_1,\theta_2,\theta_3,\ldots,\theta_K\}$. The objective of the K-means algorithm is to minimize the total distortion (or quantization error) given by:

$$D = \sum_{i=1}^{T} \sum_{j=1}^{K} \| \mathbf{x}_i - \boldsymbol{\theta}_j \|^2 \quad (1)$$

K-means is an iterative approach, in each successive iteration; it redistributes the vectors in order to minimize the distortion D (quantization error). It takes place in following steps:

Step-1 Choose arbitrary initial estimates $\theta_j(0)$ for the centroid vectors θ_j' s, j=1,2,….,K. Calculate the initial value of the distortion D(0).

Step 2: For i=1 to T

For a vector $x_{i.}$, determine the nearest centroid , say θ_j, Set centroid(i)=j (centroid or cluster for the j^{th} vector)

End

For i=1 to K

Calculate new centroids θ_j as the mean of the vectors $x_i \varepsilon X$ with centroid(i)=j.

Calculate the distortion value D(i).

End

Step 3: Repeat Step 2 until either a maximum number of iterations is reached or the distortion value D(i) falls below a preset threshold or until no change in θ_j's occurs between a few successive iterations.

The above algorithm iteratively moves the cluster boundaries. When the distortion D is minimized, subsequent iterations do not result in any movement of vectors between clusters and the cluster boundaries

become stabilized. This could be used as one of possible indicators to terminate the algorithm. The total distortion can also be used as an indicator of convergence of the algorithm. Upon convergence, the total distortion does not change as a result of redistribution. A great advantage of this algorithm is its computational simplicity.

In case of speaker recognition the speech files are preprocessed and a set of feature vectors is calculated. The K-means clustering can be then used to group feature vectors for each speaker into K sets (clusters) which efficiently describe the acoustic attributes of a given speaker. Thus, each speaker is modeled by a set of K clusters of feature vectors. An example of K-means procedure is illustrated in Fig. 2. K-means encounters the problem of overfitting [11]. The drawback of overfitting can be largely eliminated by using the ITVQ which works on the principle of physical interpretation of the data clusters.

3. LINDE BUZO GRAY CLUSTERING

The LBG algorithm [5] consists of a sequence of iterative steps minimizing the distortion measure. It consists of two phases, which are codebook initialization and codebook optimization.

The codebook optimization process is guided by minimization of the average distortion of the MQE (Maximum Quantization Error) given as:

$$MQE = D(\mathbf{X}, \mathbf{Y}, q) = \frac{1}{N} \sum_{i=1}^{N} d(\mathbf{x}_i, q(\mathbf{x}_i)) \quad (2)$$

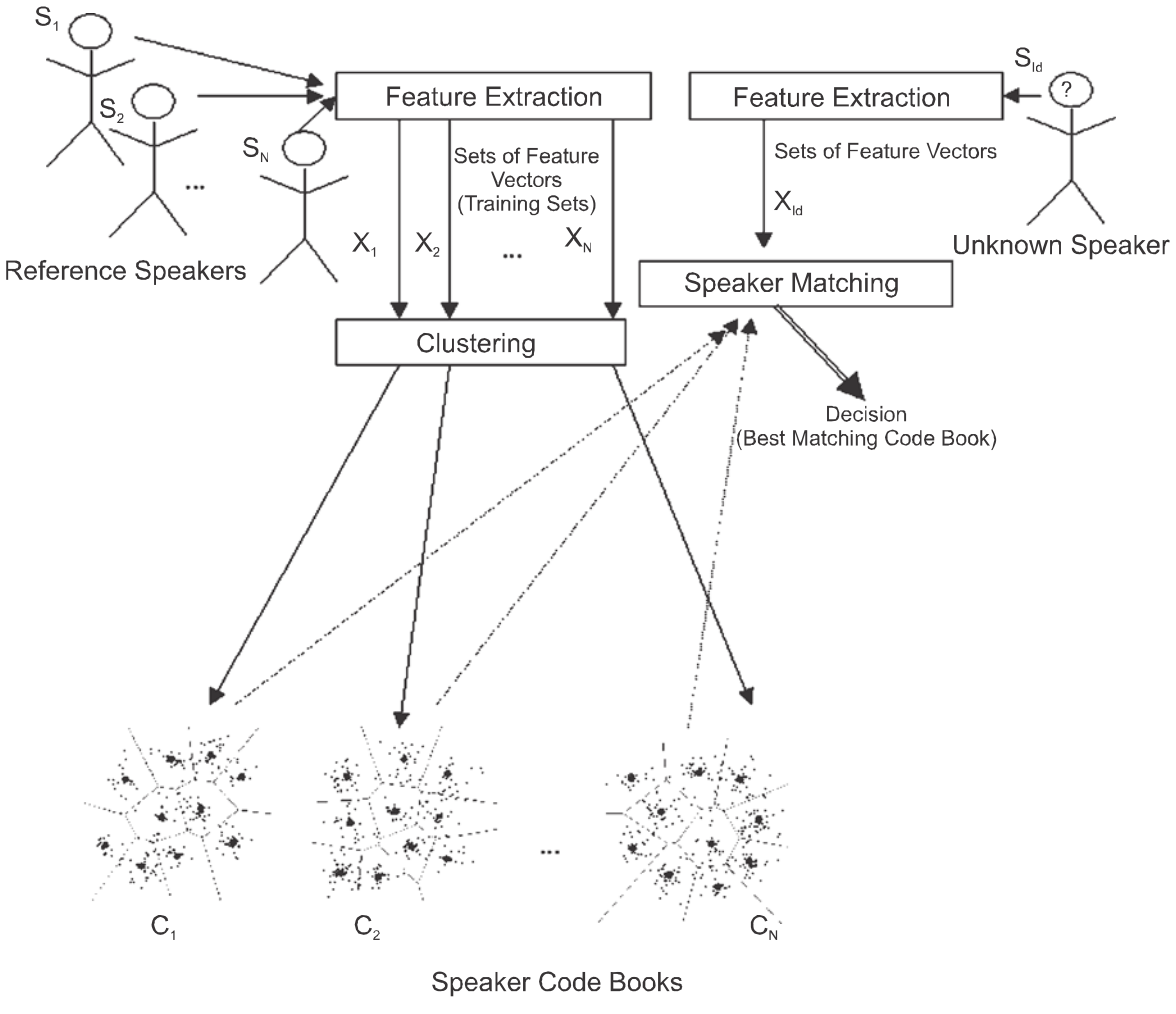

FIG. 1. STRUCTURE OF THE VQ BASED SPEAKER VERIFICATION SYSTEM.. COURTESY OF TOMI K

where Y is a given codebook, $X=\{x_1, x_2, ..., x_N\}$ is the set of observation data vectors, N is the total number of observation data vectors, d is a vector distance measure, and q is the vector quantizer function, defined such that $q(x_i)$ is the codeword assigned to vector x_i based on the nearest neighbor criterion.

The LBG algorithm requires the user to provide an initial estimate of the codebook and to specify the desired number of clusters. Due to the nature of the classical LBG algorithm, which usually generates the initial codebook by randomly splitting codewords into two new codewords, the desired number of clusters needs to be a power of 2. The following sections describe the subsequent phases of the LBG algorithm.

3.1 Codebook Initialization

The choice of initial codebook can be critical for the quality of the final solution. The poor choice of the initial codebook will lead to a final quantizer with a relatively large value of the quantization error. A number of methods such as random initialization [12], initialization by splitting [5] and maximum distance initialization [13] have been proposed to perform codebook initialization. One of the most often used approaches is based on random splitting codewords until a desired codebook size is reached. As illustrated in Fig. 3, the process of generating an initial codebook starts with a single random initial codeword. The single codeword is then randomly split into two codewords by a small random perturbation. The procedure proceeds until a pre-set number of codewords is reached. This type of codebook initialization results in a codebook size which is of power 2.

3.2 Codebook Optimization

The initialization step is followed by the iterative codebook optimization procedure which gradually improves the codebook estimate by minimization of the total distortion (quantization error) D. The optimization phase of the LBG algorithm proceeds as follows [14]:

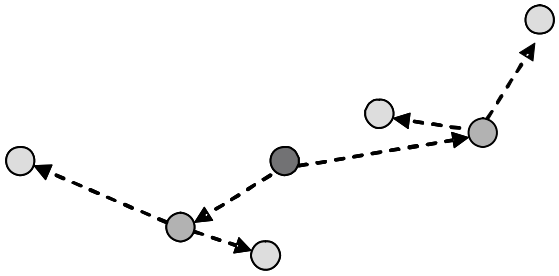

FIG. 3. INITIAL CODEBOOK GENERATION BY RANDOMLY SPLITTING THE CODEWORDS

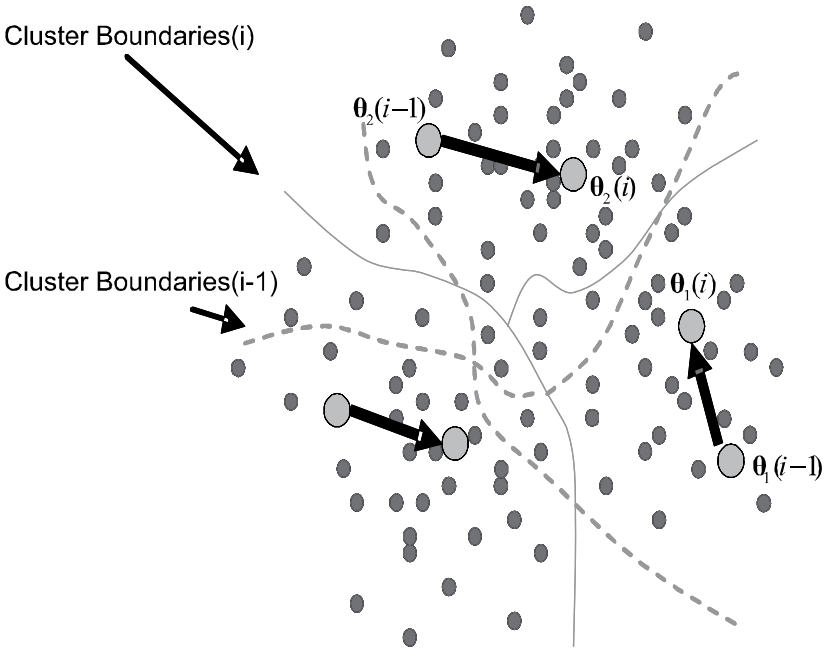

FIG. 2. K-MEANS CLUSTERING: THE SMALL DOTS REPRESENT DATA VECTORS, I IS THE ITERATION NUMBER AND J DENOTE CENTROID VECTORS (LARGE DOTS). THE LINES REPRESENT BOUNDARIES BETWEEN CLUSTERS

Step-1 Assign the initial codebook as the current codebook Yk and the current iteration number k=1.

Step-2 Using the current vector quantizer q^k, divide the training data into a set of NN clusters (also called the Voronoi clusters [14]). Then calculate the average distortion $D(Y^k,q^k)$ using Equation (2).

If $abs(D(Y^k,q^k) - D(Y^{k-p},q^{k-p}))$ is less than a preset threshold ζ, then terminate the algorithm.

else, go to Step-3. The p value is a control step denoting small number of iterations.

Step-3 Set k=k+1, and update the codebook Y^k by calculating the centroids of the new clusters, update the nearest neighbour quantizer q^k and go to Step-2.

The cycle of iterations usually continues until the decrement in average distortion value calculated over a specific small number of iterations falls below a pre-set threshold ζ. Alternatively the algorithm can be terminated when a pre-set maximum of iterations is reached.

The LBG algorithm offers a constructive solution to a very complex problem of generating an optimal VQ codebook. The advantage of the LBG is that it does not require knowledge about the underlying statistics of the observation data. However, the quality of the final solutions depends on the quality of the initial codebook. The procedure has a gradient descent character and has no mechanisms allowing escaping from local minima, therefore the algorithm has a tendency to end up in low quality local minima.

4. INFORMATION THEORETIC VECTOR QUANTIZATION

In K-means and LBG algorithms, data points are associated with the nearest code vector reducing the size of the original data. The challenge is to find the set of code vectors (the codebook) that reduces the data to a smaller set preserving the distribution of the original data vectors. Unlike the K-means or LBG, ITVQ [15] has a clear physical interpretation and relies on minimization of a well defined cost function. The ITVQ uses descriptors from information theory (entropy and divergences) estimated directly from the data to substitute the conventional statistical descriptors of variance and covariance. The ITVQ is based on a number of core concepts of the information theory such as Parzen density estimator, Kullback Leibler divergence, Cauchy Schwartz Inequality and Renyi's Quadratic Entropy [15]. In information theory minimization, the free distance between the codeword's distribution and the original data distribution is equivalent to the minimization of the divergence measure between these two distributions. The divergence measure is calculated directly from the data using the Parzen density estimator. The divergence minimization algorithm can be also seen as a probability density matching method, where the distance between the Parzen density estimator for the codewords and the Parzen density estimator for the original data is minimized. The potential field created by a single vector (particle) can be described by a kernel of the form $K(\cdot)$. Placing a kernel on each particle, the potential energy at a point in space x is given by:

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{K}\left(\mathbf{x}-\mathbf{x}_i\right) \tag{3}$$

where x_i are the data vectors. Equation (3) is known as the Parzen density estimator [16].

In order to match the distribution of the codewords with the distribution of the original data, Equation (5) can be used to estimate their densities and then minimize the divergence between the densities. The distribution of the data points (x_i) can be written as:

$$f(\mathbf{x}) = \sum_{i} G\left(\mathbf{x}-\mathbf{x}_i,\sigma_f\right) \tag{4}$$

Similarly, the distribution over codewords (w_i) can be written as:

$$g(\mathbf{x}) = \sum_{i} G\left(\mathbf{x}-\mathbf{w}_i,\sigma_g\right) \tag{5}$$

Where, G(.) represents the Gaussian kernel given as:

$$G(\mathbf{x},\sigma)=\frac{1}{\left(\sqrt{2}\pi\sigma\right)^{N}}e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}} \tag{6}$$

Numerous divergence measures exist, of which the KL (Kullback Leibler) divergence is the most commonly used [17]. The integrated square error and the CS (Cauchy Schwartz) inequality are both linear approximations to the KL divergence

The KL divergence represents a measure of the difference between two probability distributions: from a true probability distribution X to an arbitrary probability distribution Y. Typically X represents data, observations, or a precise calculated probability distribution. The measure Y typically represents a theory, a model, a description or an approximation of X. For probability distributions X and Y of a discrete random variable the KL divergence of Y from X is defined as:

$$D_{KL}\left(X\|Y\right)=\sum_{i}X(i)\log\frac{X(i)}{Y(i)} \tag{7}$$

The CS inequality is a linear approximation of the KL divergence. For vectors x and y, the inequality is written as:

$$\left|\left\langle\mathbf{x},\mathbf{y}\right\rangle\right|\le\left\|\mathbf{x}\right\|\cdot\left\|\mathbf{y}\right\| \tag{8}$$

Substituting G from Equation (6) to Equations (4-5), the distribution f(x) of the data points x_i is given as:

$$f(\mathbf{x})=\sum_i G\left(\mathbf{x}-x_i,{\sigma_f}^2\right)=\frac{1}{\left(\sqrt{2}\pi\sigma_f\right)^{N}}e^{-\frac{\left|\mathbf{x}-x_i\right|^2}{2{\sigma_f}^2}} \tag{9}$$

and the distribution g(x) of the codevectors c_j. is given as:

$$g(\mathbf{x})=\sum j\,G\left(\mathbf{x}-\mathbf{c}_j,{\sigma_g}^2\right)=\frac{1}{\left(\sqrt{2}\pi\sigma_g\right)^{M}}e^{-\frac{\left|\mathbf{x}-\mathbf{c}_j\right|^2}{2{\sigma_g}^2}} \tag{10}$$

Applying the CS inequality of Equation (8) to f(x) and g(x), we have:

$$\left|\left\langle f(\mathbf{x}),g(\mathbf{x})\right\rangle\right|\le\left\|f(\mathbf{x})\right\|\cdot\left\|g(\mathbf{x})\right\| \tag{11}$$

Equation (11) becomes equality only when f(x) and g(x) are collinear. Hence, maximizing the ratio between the numerator $\left|\left\langle f(\mathbf{x}),g(\mathbf{x})\right\rangle\right|$ and the denominator $\left\|f(\mathbf{x})\right\|\cdot\left\|g(\mathbf{x})\right\|$ is equivalent to minimizing the divergence between f(x) and g(x).

To avoid the division, the logarithm can be maximized instead. This is valid since the logarithm is a monotonically increasing function. In order to minimize the divergence between the distributions f(x) and g(x) the following expression is minimized:

$$\begin{aligned}D_{C-S}\left(f(\mathbf{x}),g(\mathbf{x})\right)&=-\log\frac{\left(\int(f(\mathbf{x})g(\mathbf{x}))dx\right)^2}{\int f^2(\mathbf{x})dx\int g^2(\mathbf{x})dx}\\&=\log\int f^2(\mathbf{x})dx-2\log\int f(\mathbf{x})g(\mathbf{x})dx+\int g^2(\mathbf{x})dx\end{aligned} \tag{12}$$

In Equation (12) the first term contains the information about the interactions between the data points. The second term addresses the interaction between the data points x_i and code vectors c_j. However, the third term is containing the information about the interactions between the code vectors itself. The interaction between the data points will not lead to an improvement, however the interactions between the data points and code vectors and between the code vectors will lead to improvement. This is because the position of data points is fixed and the only random position selection and change is associated with code vectors. Therefore, first term in Equation (12) can be ignored. The cost function with respect to code vectors can therefore be written as:

$$J(\mathbf{c}) = -2\log\int f(\mathbf{x})g(\mathbf{x})dx + \int g^2(\mathbf{x})dx \quad (13)$$

The cost function J(c) is minimized with respect to the location of the code vectors c_j. When the codevectors are located such that the local minima is achieved, no effective force acts on the code vectors. Moving the code vectors in the opposite direction of the gradient will bring them to such a potential minimum. This is also known as the gradient descent method. The gradient descent method states that the derivative of Equation (13) with respect to the location of the codevectors must be calculated. For the sake of simplicity the Eqution (13) is divided into two parts. The first part is denoted by C and the second part is denoted by V.

Considering first term of Equation (13):

$$\mathbf{C} = \int f(\mathbf{x})g(\mathbf{x})dx = \frac{1}{MN}\int \sum_{i}^{N} G_f(\mathbf{x}-x_i,\sigma_f{}^2)\sum_{j}^{M} G(\mathbf{x}-\mathbf{c}_j,\sigma_g{}^2)dx \quad (14)$$

where the covariance of the Gaussian after integration is $\sigma_a{}^2=\sigma_f{}^2+\sigma_g{}^2$. M is the number of code vector kernels and N is the number of data point kernels.

The gradient update for the code vectors c_j from the above term then becomes:

$$\frac{d}{d\mathbf{c}_j}2\log\mathbf{C} = -2\frac{\Delta\mathbf{C}}{\mathbf{C}} \quad (15)$$

where ΔC denotes the derivative of C wrt code vectors, it is calculated as:

$$\Delta\mathbf{C} = -\frac{1}{MN}\sum_{i}^{N} G_f\left(\mathbf{c}_j-\mathbf{x}_i,\sigma_f\right)\sigma_f^{-1}\left(\mathbf{c}_j-\mathbf{x}_i\right) \quad (16)$$

Similarly for the second term V we have:

$$\mathbf{V} = \int g^2(\mathbf{x})dx = \frac{1}{M^2}\sum_{j}^{M}\sum_{k}^{M} G\left(\mathbf{c}_j-\mathbf{c}_k,\sqrt{2}\sigma_g\right) \quad (17)$$

The gradient update for the code vectors c_j from the second term then becomes:

$$\frac{d}{d\mathbf{c}_j}\log\mathbf{V} = \frac{\Delta\mathbf{V}}{\mathbf{V}} \quad (18)$$

where ΔV denotes the derivative of V wrt code vectors, and it is calculated as:

$$\Delta\mathbf{V} = -\frac{1}{M^2}\sum_{j}^{M} G\left(\mathbf{c}_k-\mathbf{c}_j,\sqrt{2}\sigma_b\right)\sigma_b^{-1}\left(\mathbf{c}_k-\mathbf{c}_j\right) \quad (19)$$

where k denotes the current centroid for which the update is obtained. By substituting the simplification of the above two terms obtained in Equation (16) and Equation (19) to Equation (13), the update formula for the ITVQ can be established as:

$$\mathbf{c}_k(n+1) = \mathbf{c}_\mathbf{k}(n) - \eta\left(\frac{\Delta\mathbf{V}}{\mathbf{V}} - 2\frac{\Delta\mathbf{C}}{\mathbf{C}}\right) \quad (20)$$

where η is the step size, the ITVQ consists of n updates for each of the codevector c_k.

5. EXPERIMENTS COMPARING SPEAKER VERIFICATION BASED ON K-MEANS, LBG AND ITVQ METHODS

5.1 Overview of Speaker Verification System

The overview of the speaker verification system including training and testing phases [18] is given in Fig. 4. The speaker verification system shown in in Fig. 4 can operate in one of the two possible modes:

- The target speaker enrollment (training), and
- The testing mode.

For both of the system modes identical speech detection and feature extraction methods are used. The pre-processing method followed the SAD (Speech Activity Detection) procedure introduced by Reynolds in [19]. The voiced/silence interval were detected using an energy threshold. Previous research [20] have shown

that MFCC (Mel Frequency Cepstral Coefficients) based system is relatively robust to the changes in frame size (in the range of 20-50ms) and frame step (in the range of 1/6 to 1/3 of the frame size). Thus we employed MFCC to characterize the speaker information. The feature vector representing a given frame is a 36 dimensional vector including: 12 MFCC parameters, 12 delta parameters ΔMFCC (first derivative of MFCC), and 12 double delta parameters ΔΔMFCC (second derivative of MFCC). As illustrated in Fig. 5, the MFCC parameters were calculated by mapping the voiced speech spectrum into Mel frequency scale. This Mel frequency mapping was done by multiplying the magnitude of speech spectrum for a preprocessed frame by magnitude of triangular filters in Mel filterbank followed by log-compression of sub-band energies of the Mel-scale filters and finally DCT (Discrete Cosine Transform).

5.2 Speech Corpora

The speaker verification experiments were performed using two speech corpora: TIMIT and NIST 2004.

The TIMIT corpus was used to obtain speech samples of 630 speakers (438 male and 192 female). The recordings were made in a sound booth using fixed-text sentences read by speakers and recorded over a fixed wideband channel. The speakers used American English. The TIMIT corpora had a low environmental value since the clean wideband speech has an ideal character and does not simulate the real world conditions [21].

In order to provide a speech corpora that provides a better representation of the real life conditions, NIST 2004 was used with 616 speakers (248 male and 368 female) recorded in different environmental conditions. The recordings include conversational speech which is recorded mostly over a telephone line. For each speaker approximately 5

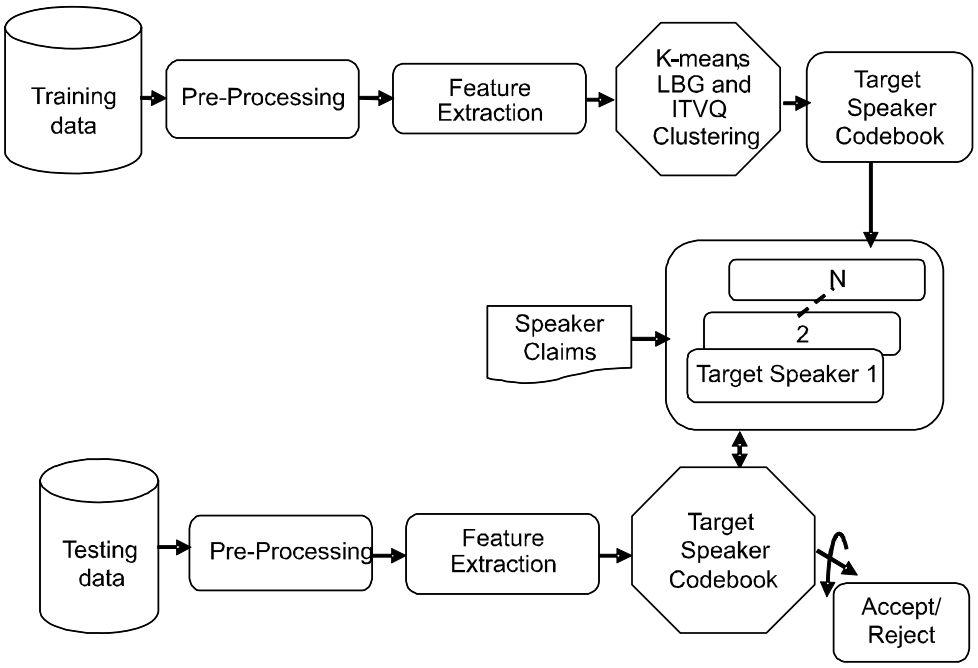

FIG. 4. BLOCK DIAGRAM OF THE SPEAKER VERIFICATION SYSTEM

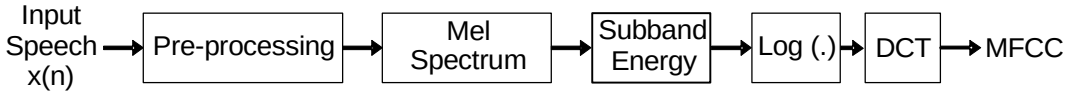

FIG. 5. CALCULATION OF THE MFCC COEFFICIENTS

minutes of speech was available for training as well as for testing. Most of the training data is in American English [22].

Table 1 shows a summary of TIMIT and NIST 2004 corpora used to perform the experiments.

5.3 Speaker Verification Results

The performance of the VQ methods is evaluated using speaker recognition rates, EER values and MSE with respect to codebooks. The speaker recognition rate was the most widely used measure to evaluate the performance of a speaker verification system. However EER measure gives a more suitable tool for the evaluation of the performance of detection systems in general and speaker verification systems in particular [22-23].

The speaker recognition rates based on different VQ methods are summarized in Fig. 6(a-b). Fig. 6(a) shows the results based on the TIMIT corpora and Fig. 6(b) shows the results based on the NIST 2004 corpora.

The recognition rates in Fig. 6 indicate that the ITVQ outperforms the conventional Kmeans and LBG methods. It also indicates that for all three algorithms, an increase of the number of clusters generally leads to a noticeable increase of recognition rates when the number of cluster increases from 32-128, further increase from 128-512 clusters shows a small degradation in performance leading to slightly lower recognition rates. The reason for the performance degradation is observed due to the increase in number of codewords which can be attributed to thinner distribution of data. With the increasing number of codewords the data is highly distributed and the codewords are therefore not capable of modeling a particular speaker accurately, which ultimately deteriorates the performance. The relatively high recognition rates for the ITVQ indicate that Parzen density estimation provides better representation of the data distribution than the mean values used in K-means and LBG algorithms. The C-S divergence minimizes the free distance between the data points and the code vectors more efficiently than the K-means and LBG methods.

TABLE 1 PROPERTIES OF THE SPEECH CORPORA

Description	TIMIT	NIST 2004
Language	English	English
Client Speakers	630	616
Speech Type	Read	Conversational
Record Condition	Lab	Telephone
Handset Mismatch	No.	No.
Sampling Rate	8KHz	8KHz
Quantization	16 Bit	8 Bit μ-Law
Train Speech	45 Second	5 Minute
Test Speech	12 Second	50 Sec

EER is used to compare the performance of speaker verification based on different VQ algorithms, since recently it has turned out to be the most widely used technique for evaluating performance of speaker recognition systems. Since the EER can only be calculated for a fixed number of codewords, a codebook containing 512 codewords was used to illustrate the performance comparison between K-means, LBG and ITVQ algorithms. Fig. 7(a-b) illustrate the percentage miss probability versus the percentage of false alarm probability and the EER values for the Kmeans, LBG and ITVQ methods using codebook size of 512. Fig. 7(a) shows the results for the TIMIT corpora and Fig. 7(b) shows the results for the NIST 2004 corpora. The miss probability measures the percent of invalid matches and the false alarm probability measures the percent of valid inputs being rejected. The EER parameter represents the rate at which both the miss and false alarm probabilities are equal. The lower the EER, the more accurate the system is considered. As illustrated in Fig. 7(a-b), both corpora show the same trend with ITVQ outperforming both K-means and LBG algorithm. The K-means algorithm provided the highest EER (34.9% for TIMIT and 21% for NIST 2004), LBG gave better performance than Kmeans (27.8% for TIMIT and 19.1% for NIST 2004). The ITVQ provided the lowest EER (15.8% for TIMIT and 11.8% for NIST 2004). The average improvement of EER value for ITVQ method is about 19.1% over K-means and 7.1% over LBG for TIMIT corpus and 9.2% over K-means and 7.3% over LBG for NIST 2004 corpus.

The MSE is calculated using the objective function for each of the evaluated procedures. Fig. 8(a) shows the MSE values based on the TIMIT corpora and Fig. 8(b) shows the MSE based on the NIST 2004 corpora. Fig.8(a-b) show the same trends as previously indicated by

classification rates and EER. The ITVQ provides the lowest MSE values and the fastest convergence rates. The LBG algorithm gives the medium performance and the K-means algorithm shows the largest MSE values and the slowest algorithm convergence rates.

6. CONCLUSIONS

In this paper we evaluated and compared the performance of mean based and information theoretic learning based vector quantization modelling techniques for the speaker verification system. The performance was compared using a feature set containing 12 MFCC with its delta and double coefficients. The evaluation was based on two speech corpora: TIMIT and NIST 2004. The results were evaluated in terms of three different performance measures: classification rates, EER and MSE.

The results based on these different classifiers and speech corpora were consistent indicating that the vector quantization based on IT learning established better recognition rates. The better performance of the ITVQ algorithm can be attributed to the fact that the IT criteria used by this algorithm provide better matching between distribution of the original data vectors and the codewords.

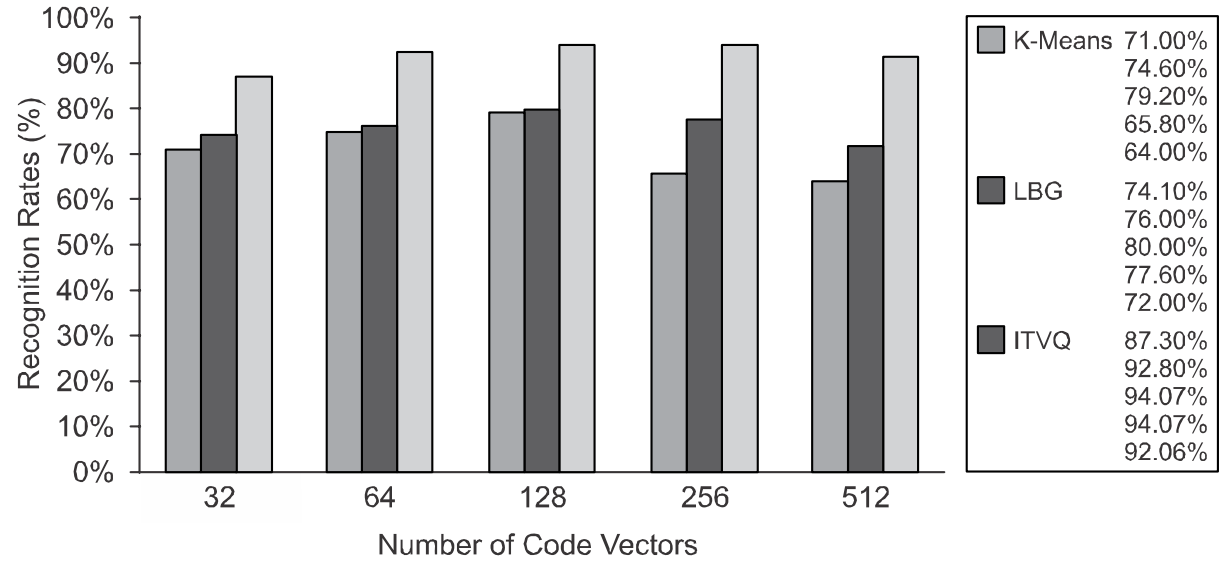

FIG. 6(A). RECOGNITION SCORES FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR TIMIT SPEECH CORPORA.

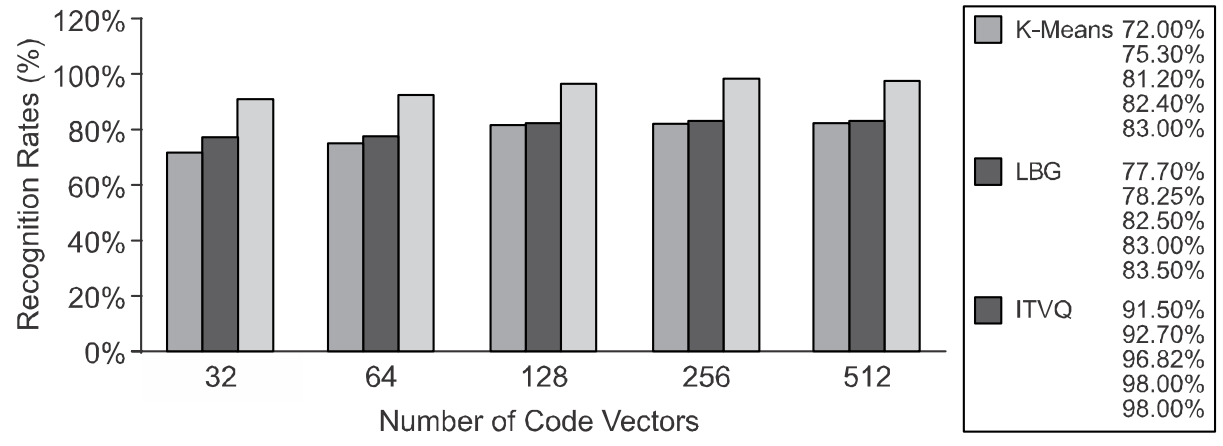

FIG. 6(B). RECOGNITION SCORES FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR NIST'04 SPEECH CORPORA

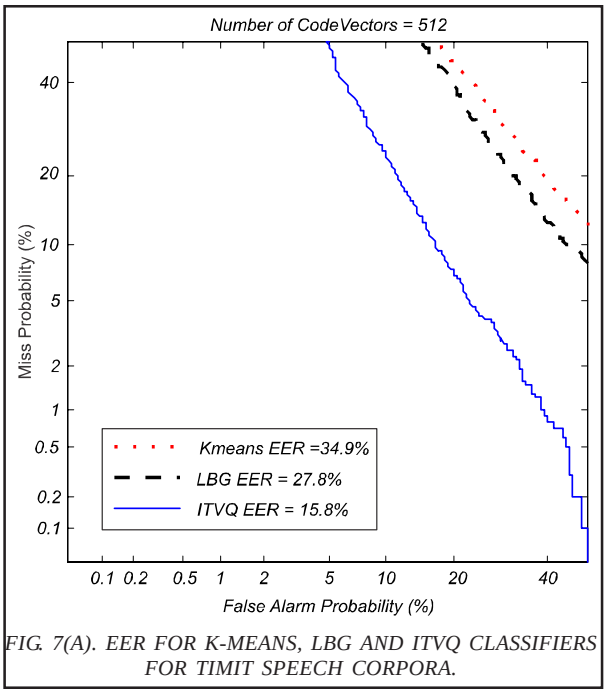

FIG. 7(A). EER FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR TIMIT SPEECH CORPORA.

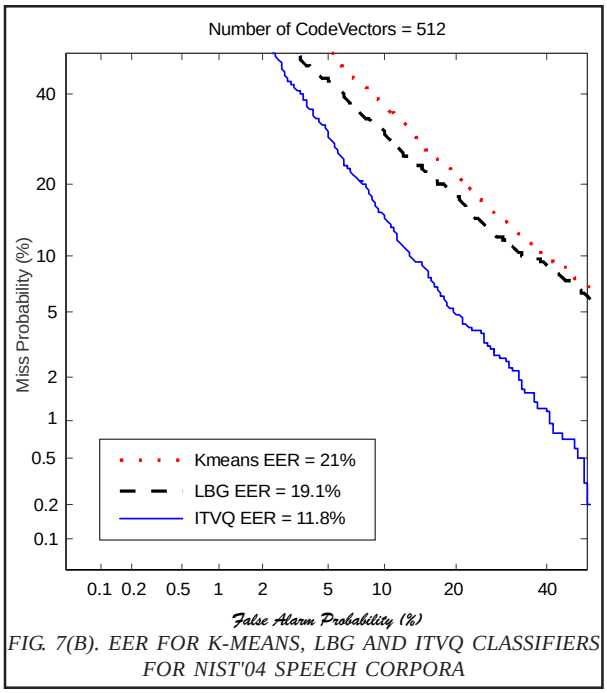

FIG. 7(B). EER FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR NIST'04 SPEECH CORPORA

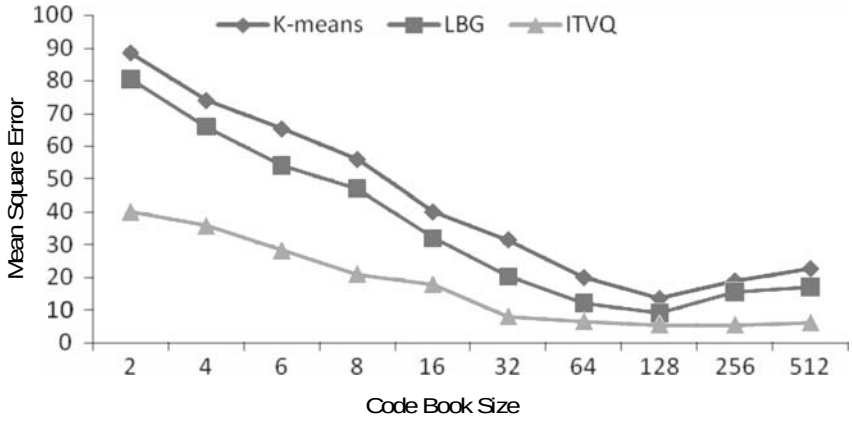

FIG. 8(A). MSE FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR TIMIT SPEECH CORPORA.

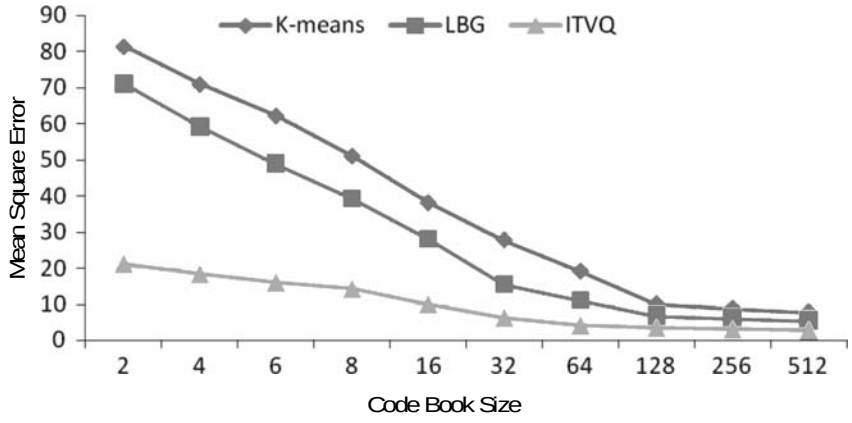

FIG. 8(B). MSE FOR K-MEANS, LBG AND ITVQ CLASSIFIERS FOR NIST'04 SPEECH CORPORA

ACKNOWLEDGEMENT

This work would not have been possible without the encouragement and support of Prof. Dr. Abdul Qadir Khan Rajput, Vice-Chancellor, Prof. Dr. Bhawani Shankar Chowdhry, Dean, Faculty of Electrical, Electronic & Computer System Engineering, and Prof. Dr. Mukhtiar Ali Unar, Director, Institute of Information & Communication Technologies, Mehran University of Engineering & Technology, Jamshoro, Pakistan.

REFERENCES

[1] Gray, R., "Vector Quantization", IEEE Magazine on Acoustics Speech and Signal Processing, Volume 1, pp. 4-29, April, 1984.

[2] Matsui, T., and Furui, S., "Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMMs", IEEE Transactions on Speech Audio Processing, Volume 2, No. 3, pp. 456-9, July, 1994.

[3] Furui, S., "Vector Quantization Based Speech Recognition and Speaker Recognition Techniques", 25th Asilmor Conference Signals, Systems and Computers, 1991.

[4] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations", Fifth Berkeley Symposium on Mathematical Statistics and Probability Volume 1, pp. 281-297, 1967.

[5] Linde, Y., Buzo, A., and Gray, R.M., "An Algorithm for Vector Quantizer Design", IEEE Transactions on Communications, Volume 28, No. 1, pp. 84-95, 1980.

[6] Inal, M., and Fatihoglu, Y.S., "Self Organizing Map and Associative Memory Model Hybrid Classifier for Speaker Recognition", Seminar on Application of Neural Netwroks in Electrical Engineering, pp. 71-4, Belgrade, Yugoslavia, September, 2002.

[7] Kinnunen, T., Kilpeläinen, T., and Fränti, P., "Comparison of Clustering Algorithms in Speaker Identification", Proceedings of the IASTED International Conference on Signal Processing and Communications, pp. 222-227, Marbella, Spain, Septmber, 2000.

[8] Memon, S., Lech, M., "Speaker Verification Based on Information Theoretic Vector Quantization", Communications in Computer and Information Science, Wireless Networks, Information Processing and Systems, Springer Berlin Heidelberg 2009, pp. 391-399. Vol. 56, No. 7, pp. 2797-2811, July 2008.

[9] Memon, S., and Lech, M., "Using Information Theoretic Vector Quantization for GMM Based Speaker Verification", 16th European Signal Processing Conference, Lausanne, Switzerland, August 25-29, 2008.

[10] Furui, S., "Digital Speech Processing, Synthesis and Recognition", Marcel Dekker Inc., New York, 1989.

[11] Gresho, A., "Vector Quantization and Signal Compression", Kulwer Academic Publishers, 1992.

[12] Pal, N., Bezdek, J., and Tsao, E., "Generalized Clustering Networks on Kohonen's Self Organizing Scheme", IEEE Transactions on Neural Networks Volume 4, pp. 549-557, 1993.

[13] Katsavounidis, I., Kuo, C.C., and Zhang, Z., "A New Initialization Technique for Generalized Lloyd Iteration", IEEE Signal Processing Letters, Volume 1, pp. 144-146, 1994.

[14] Lech, M., "Algorithms for the Vector Quantization of Images", Ph.D. Thesis, The University of Melbourne, 1993.

[15] Tue, L.S., Anant, H., Deniz, E., and Jose, C.P., "Vector Quantization Using Information Theoretic Concepts", Natural Computing, Volume 4, pp. 39-51, Springer, 2005.

[16] Parzen, E., "On Estimation of a Probability Density Function and Mode", The Annals of Mathematical Statistics, Volume 27, pp. 1065-1076, 1962.

[17] Kullback, S., and Leibler, R.A., "On Information and Sufficiency", The Annals of Mathematical Statistics, Volume 22, pp. 79-86, 1951.

[18] Ganchev, T.D., "Speaker Recognition", Ph.D. Dissertation, University of Patras, Greece, 2005.

[19] Reynolds, D.A., Rose, R.C., and Smith, M.J.T., "PC-Based TMS320C30 Implementation of the Gaussian Mixture Model Text-Independent Speaker Recognition System", Proceedings of the International Conference on Signal Processing Applications and Technology, pp. 967-973, November, 1992.

[20] Reynolds, D.A., "Experimental Evaluation of Features for Robust Speaker Identification", IEEE Transactions on Speech Audio Process, Volume 2, No. 4, pp. 639-643, October,. 1994.

[21] John, S.G., Lori, F.L., William, M.F., Jonathan, G.F., David, S.P., Nancy, I.D., and Victor, Z., "TIMIT Acoustic-Phonetic Continuous Speech Corpus", Linguistic Data Consortium, 1993.

[22] NIST "Speaker Recognition Evaluation", 2004. http://www.itl.nist.gov/iad/mig/tests/spk/2004/.

[23] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybock, M., "The DET Curve in Assessment of Detection Task Performance", EUROSPEECH, pp.1895-1898, 1997.