16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

USING INFORMATION THEORETIC VECTOR QUANTIZATION FOR GMM BASED SPEAKER VERIFICATION

Sheeraz Memon; and Margaret Lech

School of Electrical and Computer Engineering,
RMIT University, Melbourne, VIC, 3001, Australia
Phone: +61 3431284176; Fax: +61 3 9925 2007;
email: sheeraz.memon@student.rmit.edu.au, margaret.lech@rmit.edu.au

ABSTRACT

The introduction of Gaussian mixture models in the field of voice recognition systems has established very good results. The process of speaker verification based on Gaussian mixture models is highly expensive in the regard of computational complexity and memory usage perspectives, thus suppressing its adaptability for efficient and low-cost systems. The methods like Expectation Maximization used by GMM to compute the speaker models are highly iterative procedures and contribute significantly to the complexity in the implementation of an efficient system. In this paper we propose the use of Information theoretic vector quantization VQIT for the training of GMM models as a replacement of EM algorithm; we also apply the other vector quantization techniques such as K-means and LBG and compare the performance with the VQIT.

1. INTRODUCTION

Speaker verification is a process where a person is identified or verified biometrically, for such systems a person's identity is verified by voice. Using the voice of a speaker to identify a person has become the more applied approach in the recent years because as such of the characteristics do not need to be memorised like passwords. The favourite part of the speaker verification systems is that the speaker is not bound to speak any restricted phrase to get identified but he is free to utter any sentence. The speaker verification system comprises of three stages, in the first stage feature extraction is performed over a database of speakers where vectors representing the speaker distinguishing characteristics are isolated. The second step addresses establishing the speaker model; this translates to finding the distribution of feature vectors. The third step is of decision, which determines the claimed identity of a speaker, as depicted in fig.1.

Vector quantization (VQ) based speaker verification has remained a successful method in the field of speaker recognition systems**.** The basic idea in this approach is to compress a large number of short term spectral vectors into a small set of code vectors. Until the evolution of GMM, vector quantization techniques were the largely applied in the field of speaker verification, and in this paper we come up with the idea that using vector quantization as part of a GMM based speaker verification can serve in terms of better results. The fundamental problem with the GMM inspite of having very high recognition rates is its computational complexity, efficiency and its adaptability for the low cost systems. The GMM uses the EM algorithm to train the speaker models which actually contributes to the complexity of system. We suggest that if we use the vector quantization techniques [1], [2] instead of EM algorithm then it not only reduces its computational complexity but also adds to its adaptability for low-cost systems. The major contribution of our work appears when we use the information theoretic vector quantization VQIT [4] along with other VQ implementations like LBG and K-means where VQIT establishes good results.

2. PRELIMINARIES

In this section we shall discuss the K-means, LBG and VQIT algorithms for vector quantization and their application as individual classifiers to speaker verification. Further we will discuss the GMM algorithm and its EM architecture.

2.1 K-means algorithm

It is an algorithm to classify or to group data based on attributes/features into K number of group. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of K-mean clustering is to classify the data. K-means algorithm [5] was developed for vector quantization codebook generation. It represents each cluster by the mean of the cluster. Assume a set of vectors $X=\{x_1,x_2,x_3,\ldots..,x_T\}$ is to be divided into M clusters represented by their mean vectors $\{\mu_1, \mu_2, \mu_3,\ldots, \mu_M\}$ the objective of K-means algorithm is to minimize the total distortion given by,

$$total\_distortion = \sum_{i=1}^{M} \sum_{t=1}^{T} \left\| x_t - \mu_i \right\| \qquad (1)$$

K-means is an iterative approach; in each successive iteration it redistributes the vectors in order to minimize the distortion. The procedure is outlined below:

(a) Initialize the randomized centroids as the means of M clusters.
(b) Data points are associated with the nearest centroid.
(c) The centroids are moved to the centre of their respective clusters.

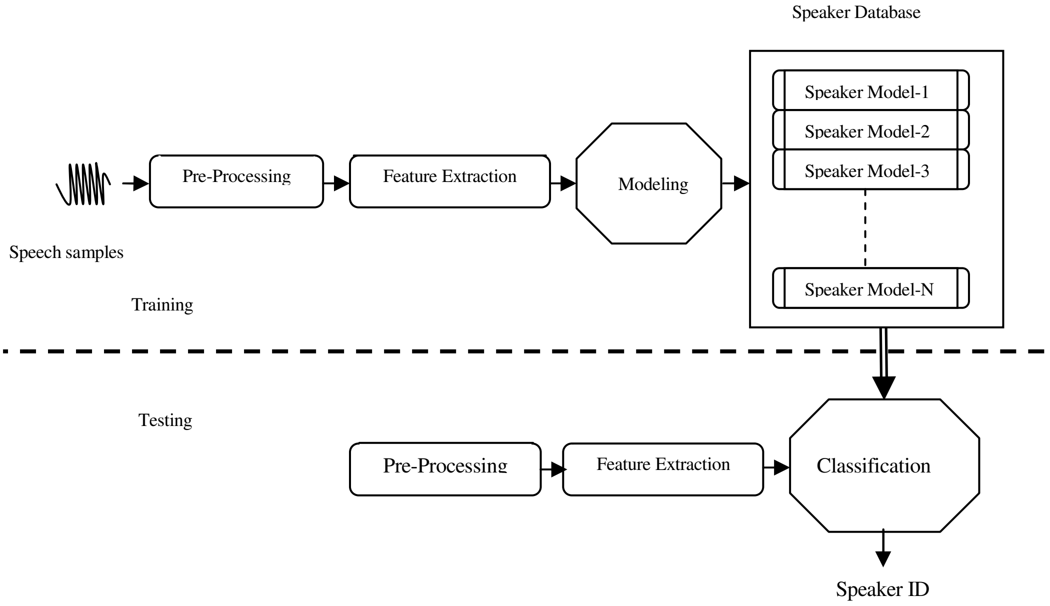

Fig.1. The three stage SV phenomenon involving feature extraction, modelling and classification

(d) Steps b & c are repeated until a suitable level of convergence has been reached, i.e. the distortion is minimized.

When the distortion is minimized, redistribution does not result in any movement of vectors among the clusters. This could be used as an indicator 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. It **is** to be noted that in each iteration, K-means estimates the means of all the *M* clusters.

2.2 LBG (Linde, Buzo and Gray) Algorithm

The LBG algorithm is a finite sequence of steps in which, at every step, a new quantizer, with a total distortion less or equal to the previous one, is produced. We can distinguish two phases, the initialization of the codebook and its optimization. The codebook optimization starts from an initial codebook and, after some iterations, generates a final codebook with a distortion corresponding to a local minimum. The following are the steps for LBG algorithm.

a. Initialization. The following values are fixed:

- N_C: number of codewords;
- $\varepsilon \geq 0$: precision of the optimization process;
- Y_0: initial codebook;
- $X = \{\mathbf{x}_j\,; j = 1\,, ..., N_P\}$: input patterns;

Further, the following assignments are made:

- $m = 0$; where m is the iteration number.
- $D_{-1} = +\infty$; where D is the minimum quantization error calculated at every mth iteration.

b. Partition calculation. Given the codebook Y_m, the partition $P(Y_m)$ is calculated according to the *nearest neighbour condition*, given by

$$S_i = \{x\varepsilon X : d(x, y_i) \geq d(x, y_j), \quad j = 1,2,....,N_C, j \neq i\} \quad i=1,2,....,N_C. \tag{2}$$

c. Termination condition check. The quantizer distortion ($D_m = D(\{Y_m, P(Ym)\})$ is calculated according to following equation.

$$MQE \equiv D(\{Y,S\}) = \frac{1}{N_P}\sum_{p=1}^{N_P} d(x_p, q(x_p)) = \frac{1}{N_P}\sum_{i=1}^{N_C} D_i \tag{3}$$

Where D$_i$ indicates the total distortion of ith cell.

If $|(D_{m-1} - D_m)| / D_m \leq \varepsilon$ then the optimization ends and Y_m is the final returned codebook.

d. New codebook calculation. Given the partition $P(Y_m)$, the new codebook is calculated according to the Centroid condition. In symbols:

$$Y_{m+1} = X\,(P(Y_m)) \tag{4}$$

After, the counter m is increased by one and the procedure follows from step b.

2.3 VQIT (Information theoretic vector quantization) algorithm

In Vector quantization the challenge is to find a way that best represents the data. VQIT [4] uses a new set of concepts from information theory and eliminates the flaws of previous vector quantization algorithms. Unlike LBG and Self Organizing map (SOM) algorithms this algorithm addresses a clear physical interpretation of data and relies on minimization of a well defined cost function. In the light of information theory it becomes clear that minimizing distance is actually equivalent to minimizing the divergence between distribution of data and distribution of code vectors. When SOM is converged it is at the minimum of cost function, but this cost function is highly discontinuous and drastically changes if any sample changes its best matching centroids [6]. Now attempts have been made to find a cost function that when minimized gives results similar to the original update rule [7] and information theorists have made attempts to design good vector quantifiers [8], [9] and [10]. Unlikely the [7], [8], [9] and [10] VQIT takes the distribution of data explicitly into account by matching the distribution of the code vectors with the distribution of the data points in a data cluster. This approach leads to the minimization of the well defined cost function.

This algorithm works on the principal of minimizing the divergence between Parzen estimator of the code vectors density distributions and a Parzen estimator of the data distribution. Minimizing the divergence between the Parzen estimates of data points and code vectors means minimizing the dissimilarity, and this is achieved by using Cauchy-Schwartz inequality which is the linear approximation of kullback-leibler divergence.

The Parzen density estimator is given by the following equation,

$$p(x)=\frac{1}{N}\sum_{i=1}^{N}K(x-x_i) \qquad (5)$$

Where K(.) is the Gaussian Kernel and x is the independent variable for which we seek the estimate and x_i represents the data points. The Parzen estimate of the data has N kernels, where N is the number of data points and the Parzen estimator of the code vectors has the M kernels, where M is the number of code vectors and M<<N. After evaluating the density estimation the divergence measure is evaluated and this is achieved by using Cauchy-Schwartz inequality,

$$|a(x)\ b(x)| \leq \|a(x)\|\ \|b(x)\| \qquad (6a)$$

This equation is used to minimize the divergence between a(x) and b(x), where a and b represent the data points and code vectors respectively. Hence Maximizing the following expression is equivalent to minimizing divergence between a and b.

$$\frac{|a(x)b(x)|}{\|a(x)\|\|b(x)\|} \qquad (6b)$$

In order to minimize the divergence between data points (Say a(x)) and Code Vectors (Say b(x)) the following is minimized.

$$D_{C-S}(a(x),b(x))=-\log\frac{\left(\int(a(x)b(x))dx\right)^2}{\int a^2(x)dx\int b^2(x)dx} \qquad (7)$$

The cost function used to evaluate the centroids for code vectors can further be achieved as,

$$J(w)=\log\int a^2(x)dx-2\log\int a(x)b(x)dx+\int b^2(x)dx \qquad (8)$$

This cost function is minimized with respect to the location of the centroids (w). When the centroids reach to a location so that a local minimum is achieved then no effective force acts on them, this uses gradient descent method for addressing the local minima. Eq. (8) has three terms the first term represent the data points which are stationary thus differentiating Eq.8 with respect to centroids will yield zero for the first term, the middle term called cross information potential and the last term called the entropy of centroids [4] will have non-zero derivatives.

Consider the cross information potential term; the Parzen estimator for a(x) and b(x) puts Gaussian kernels on each data point x_j and each centroid w_i respectively, where the variances of the kernels are σ_a^2 and σ_b^2. Initially the locations of the centroid are chosen randomly.

$$C=\int a(x)b(x)dx \qquad (9a)$$

$$=\frac{1}{MN}\int\sum_{i}^{M}G(x-w_i,\sigma_b^2)\sum_{j}^{N}G(x-x_j,\sigma_a^2)dx \qquad (9b)$$

$$=\frac{1}{MN}\sum_{i}^{M}\ \sum_{j}^{N}G(w_i-x_j,\sigma_f^2),\ \text{and}\ \sigma_f^2=\sigma_a^2+\sigma_b^2 \qquad (9c)$$

Where M represents the number of centroid kernels and N represents the number of data point kernels. The gradient update for the centroid w_k from the cross information potential term then becomes,

$$\frac{d}{dw_k}2\log C=-2\frac{\Delta C}{C} \qquad (10)$$

Where ΔC denotes the derivative of C w.r.t w_k, and ΔC is calculated as,

$$\Delta C=-\frac{1}{MN}\sum_{j}^{N}G_f(w_k-x_j,\sigma_f)\sigma_f^{-1}(w_k-x_j) \qquad (11)$$

Similarly for the entropy term we have,

$$V=\int b^2(x)dx=\frac{1}{M^2}\sum_{i}^{M}\ \sum_{j}^{M}G(w_i-w_j,\sqrt{2}\sigma_b) \qquad (12a)$$

$$\frac{d}{dw_k}\log V=\frac{\Delta V}{V} \qquad (12b)$$

With,

$$\Delta V=-\frac{1}{M^2}\sum_{i}^{M}G(w_k-w_i,\sqrt{2}\sigma_b)\sigma_b^{-1}(w_k-w_i) \qquad (13)$$

The update for point k consist of two terms, cross information potential and entropy of the centroids,

$$w_k(n+1)=w_k(n)-\eta\left(\frac{\Delta V}{V}-2\frac{\Delta C}{C}\right) \qquad (14)$$

Where η is the step size, the VQIT consists of a loop over all w_k.

2.4 Gaussian Mixture Models

Gaussian Mixture Model (GMM) has appeared as a widely applied tool in the field of text-independent speaker verification [12] in recent years. The discouraging attribute of GMM is training of GMM models, when we approach to design a speaker model by giving its feature vectors; it leads to complexity and low-efficiency due to its computational complexity and iterative nature. The iterative procedure and training of a GMM model utilizes an algorithm called EM algorithm**,** The EM algorithm achieves the convergence to a local maximum. However, the high computational complexity of the algorithm necessitates high hardware cost **as** well as large training time. In this paper, we have proposed a vector quantization algorithm called VQIT and its performance supremacy to other VQ algorithms for the training of GMM speaker models.

Although EM algorithm performs well to establish the convergence of local maximum, but by adding following disadvantages:

(a) Usually the EM algorithm guarantees the convergence of local maximum after it takes 8 to 16 iterations. The iterative procedure adds to the complexity of the operations and thus reduces the efficiency.
(b) The EM algorithm performs the intensive computational operations such as square root, exponentiation and division, and in case when we have more training vectors the numbers of such operations grow exponentially.
(c) If we are dealing with low-cost systems it will be highly impossible to establish a high-speed single-chip implementation of EM algorithm.

Thus investigation of alternative training algorithms is unavoidable. The K-means algorithm has been applied for finding a robust model approximation to the GMM in [3]. Hence we are using a number of vector quantization algorithms including K-means, LBG and recently designed VQIT to investigate its suitability to avoid complexity and efficiency loses when using EM algorithm. We also compare the performance of VQIT over other vector quantization approaches which are previously applied as a replacement measure of EM algorithm [2].

3. RESULTS AND DISCUSSION

Experiments were conducted on TIMIT databases to investigate the suitability of VQIT algorithm for speaker verification. In this section the results of the experiments are presented.

3.1 Experimental Set-up:

3.1.1 Speaker Corpus

We have enrolled a set of 36 speakers of the New England dialect from the TIMIT corpus by keeping a half of male and female speakers; we are using two min and thirty seconds of speech files after removing the silence, two minutes for training and thirty seconds for testing. The TIMIT corpus of read speech has been designed to provide speaker data. The speech was recorded at Texas instrumentation (TI), transcribed at Massachusetts Institute of Technology (MIT), and has been maintained, verified, and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST).

3.1.2 Preprocessing and extracting features

The speech data is preprocessed before extracting the coefficients. We are using a high-pass filter: v (k) =x (k)-0.95x (k-1) which pre-emphasizes the speech data where x (k) contains the speech sampled data and v (k) returns the pre-emphasized speech data. We are using a logarithmic technique suggested by [11] for separating and segmenting speech from noisy background environments, a set of efficient rules is used to generate speech and noise metrics from the input speech. The rules are derived from the statistical principles about the characteristics of the speech and noise waveform and are based on time-domain processing to have zero-delay decision; at the final stage the algorithm compares the speech and silence metrics using a threshold scheme to control the speech/silence decision. Therefore silence is removed from the speech files before applying the feature extraction algorithms to pre-emphasised speech data. The feature extraction algorithms used to extract the coefficients are linear predictive coefficients (LPC) and mel-frequency Cepstral coefficients (MFCC) which generates the coefficient data ready for training.

3.2 Experimental Results

Our principal objective of carrying on this experiment was to test the vector quantization algorithm called VQIT for the speaker verification data. We carry the experiments with some other implementations of vector quantization such as K-means and LBG to make a comparative analysis and we observed that VQIT behaves a better vector quantization approach than the other VQ implementations as listed in tables 1 and 2. Table1 indicates the performance comparison of different VQ implementations when LPC is used as a feature extraction algorithm, and Table2 indicates the performance comparison when MFCC is used as a feature extraction technique. We train the coefficient data by using the different vector quantization versions along with GMM to optimize the speaker models.

Algorithm	Speech Length		Rate of verifying an speaker
	Training	Testing	
GMM_EM	2 min	30sec	81%
GMM_K-means	2 min	30sec	73%
GMM_LBG	2 min	30sec	74.5%
GMM_VQIT	2 min	30sec	79.8%

Table 1: Performance comparison of different vector quantization techniques when feature extraction algorithm is LPC

Algorithm	Speech Length		Rate of verifying an speaker
	Training	Testing	
GMM_EM	2 min	30sec	96.5%
GMM_K-means	2 min	30sec	80%
GMM_LBG	2 min	30sec	81.5%
GMM_VQIT	2 min	30sec	96%

Table 2: Performance comparison of different vector quantization techniques when feature extraction algorithm is MFCC

4. CONCLUSION

In this paper we evaluate the performance of VQIT algorithm and conclude that it proves better in performance than the other Vector Quantization algorithms, when used as a replacement of EM algorithm as a part of GMM system to optimize the speaker models. We test it with different feature extraction algorithms like the linear predictive coefficient and mel-frequency Cepstral coefficients and for both of the implementation we observe the performance superiority of VQIT algorithm, when used as an optimization algorithm for

GMM classifier. Our conclusion supports the idea that in general vector quantization techniques and in particular VQIT can be employed with the GMM based classification to suppress the complexity of the algorithm and increase the efficiency of the system.

REFERENCES

1. Jialong, H., Liu, L., Gunther, P.: A new codebook training algorithm For VQ-based speaker recognition. In: IEEE international conference on acoustics, speech and signal processing. Vol. 2, pp.1091—1094. (1997).
2. Singh, G., Panda, A., Bhattacharyya, S., Srikanthan, T.: Vector quantization techniques for GMM based speaker verification. In: IEEE international conference on acoustics, speech and signal processing. Vol. 2, pp. II65-II68. (2003).
3. Pelecanos, J., Myers, S., Sridharan, S., Chandran, V.: Vector Quantization Based Gaussian Modelling for Speaker Verification. In: International conference on pattern recognition. Vol. 3, pp. 294-297. (2000).
4. Tue, L., Anant, H., Deniz, E., Jose, C.: Vector Quantization using information theoretic concepts. In: Natural Computing: an international journal. vol . 4, Issue. 1, pp. 39 – 51. (January 2005).
5. Furui, S.: Digital Speech Processing, Synthesis and Recognition, Marcel Dekker Inc., New York, (1989).
6. Erwin, E., Obermayer, K., Schulten, K.: Selg organizing maps, ordering, convergence properties and energy functions. In: Biological Cybernetics. vol. 67, No.1, pp. 47-55.(May 1991).
7. Heskes, T., Kapen, B.: Error potentials for Self organization. In: IEEE international conference on Neural Networks. vol3, pp.1219-1223, 1993.
8. Heskes, T.: Energy functions for self organizing maps. In: Kohonen Maps, E. Oja and S. Kaski, Eds., Amsterdam. pp. 303-315, Elsevier. (1999)
9. Hulle, M.V.: Kernel based topographic map formation achieved with an information-theoretic approach. In: Neural Networks. vol. 15, Issue 8-9, pp. 1029-1039, (Oct 2002).
10. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: a principled alternative to the self-organizing map, In: Artificial neural networks – ICANN 96, International Conference proceedings pp.165-701, (July 1996).
11. Lynch, Jr., Josenhans, J., Crochiere, R.: Speech/Silence segmentation for real-time coding via rule based adaptive endpoint detection, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 12, pp.1348-1351, (April 1987).
12. Reynolds, D.A., Rose, R.C.: Robust Text-independent speaker identification using Gaussian mixture models, In: IEEE transactions on speech and audio processing, vol. 3, No. 1, pp. 72-82 (1995).