Evaluation of Hybrid Vector Quantization and Hidden Markov Model Methods in Noisy Environments

MOHD ZAIZU ILYAS, SALINA ABDUL SAMAD, AINI HUSSAIN, KHAIRUL ANUAR ISHAK & ASHRANI A. ABD. RAHNI
Department of Electrical, Electronics and Systems Engineering, Faculty of Engineering,
Universiti Kebangsaan Malaysia,
43600 UKM Bangi, Selangor,
MALAYSIA

Abstract: - In this paper, we presents a comparison between Hidden Markov Model (HMM) and an approach using a hybrid of Vector Quantization (VQ) with HMM methods. The aim of combination scheme used is to improve the standalone HMM performance. A Malay spoken digit database is used for the testing and validation modules. It is shown that, in clean environments, a total success rate (TSR) of 99.97% is achieved using this hybrid approach. For speaker verification, the true speaker rejection rate is 0.06% while the impostor acceptance rate is 0.03% and the equal error rate (EER) is 11.72%. Meanwhile, in noisy environments, TSRs of between 62.57%-76.80% are achieved for SNRs of 0-30 dBs.

Key-Words: - Speaker Verification, Speech Recognition, Vector Quantization, Hidden Markov Model

1 Introduction

Speaker recognition or verification is a biometric modality that uses an individual's voice for recognition or verification purpose. It is a different technology from speech recognition, which recognizes words as they are articulated [1]. Speech contains many characteristics that are specific to each individual. For this reason, listeners are often able to recognize the speaker's identity fairly quickly even without looking at the speaker. Speaker verification is a process of determining whether a person is who he or she claims to be by using his or her voice [1,2,3,4,5].

For many years research on speaker verification has been done and some of them have reached high performance level. Many techniques have been proposed for speaker verification systems including dynamic time wrapping (DTW), hidden Markov models (HMM), artificial neural networks (ANN) and vector quantization (VQ). Recent studies show that high performance of text dependant speaker verification can be achieved using the HMM approach [1, 2]. However, in most real world applications, the speech from speakers is captured in non-ideal situations such as in noisy environments which may seriously reduce system performance [6].

This paper presents a hybrid approach of VQ and HMM. The objective is to improve the performance of HMM in a speaker verification system for both clean and noisy environments. The technique is evaluated using Malay spoken digit database for a clean environment and Gaussian white noise is added to the data to evaluate the system performance for a noisy environment. The remaining sections of this paper are organized as follows. Section 2 describes the Malay spoken digit database. Section 3 and section 4 respectively presents the details of the VQ and HMM techniques. Experimental results are discussed in sections 5 and 6. Finally, concluding remarks are presented in section 7.

2 Malay Spoken Digit Database

The raw Malay Spoken digit database was collected at Faculty of Language and Linguistic, University Malaya as part of a Malay corpus database. The database was analyzed, processed and categorized at the Signal Processing Laboratory, Faculty of Engineering, Universiti Kebangsaan Malaysia. The Malay spoken digit database contains continuous spoken digit from 0 to 9 in slow (with silent gaps or stop) and fast (without silent gaps or stops) speech and obtained in a recording room environment. The database comprises of 212 Mb of spoken digit speech spoken by 100 speakers of different races, ages and background. The speech material is stored in WAV format with a 16-bit audio sample size and at 16 kHz audio sampling rate. Table 1 summarizes the database. Out of 100 speakers, 16 of them are males and 84 of them are females. The highest population of the speakers was Malay which represents about 49% of the population. Chinese speakers represent 35% of the population followed by Indian speakers of about 13%. The average speaker age is about 26 years old and represents more than half of the total population.

Table 1 Malay Spoken Digit Database description.

Speakers	100 (16 Male / 84 Female)
Session/Speakers	1
Type of speech	Prompted Malay digit (0 to 9)
Microphone	Standard microphone
Acoustic environment	Recording room (±55dB)
Audio sample size	16 bit
Audio sampling rate	16 KHz
File format	Wave

3 Vector Quantization

Vector quantization (VQ) is a process of mapping vectors of a large vector space to a finite number of regions in that space. Each region is called a *cluster* and is represented by its centre (called a *centroid*) [7,8]. A collection of all the centroids makes up the codebook. The amount of data is significantly less, since the number of centroids is at least ten times smaller than the number of vectors in the original sample. This will reduce the amount of computations needed for comparison in later stages. Even though the codebook is smaller than the original sample, it still accurately represents a person's voice characteristics. The only difference is that there will be some spectral distortion.

In an earlier feature extraction stage, we calculated the LPC cepstrum, and the entire speech signal is represented with the LPC cepstral parameters and a large sample of these parameters is then used as the training vectors. During the training process of VQ, a codebook is obtained from these sets of training vectors. An element in a finite set of spectra in a codebook is called a codevector. The codebooks are used to generate indices or discrete symbols that will be used by the discrete HMM. Hence, data compression of speech is accomplished by VQ in the training phase and the encoding phase in finding the best codevectors for the input vectors.

To implement VQ, we must initially get the codebook. A large set of spectral analysis vectors (or speech feature vectors) is required to form the training step. If we denote the size of the VQ codebook as $M = 2^N$ codewords, then we require an L (with L >> M) number of training vectors [7,8]. It has been found that L should at least be 10M in order to train a VQ codebook that works well. For this project, we will be using the LBG algorithm [9], also known as the binary split algorithm. The algorithm is implemented by the following recursive procedure:

1. Design a 1-vector codebook: this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook W_i according to the rule:

$$W_i^+ = W_i(1+\delta) \text{ and } W_i^- = W_i(1-\delta) \qquad (1)$$

with δ = splitting parameter

and i = 1,2,…. M.

3. Nearest – Neighbor Search: for each training vector, find the centroid in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest centroid). This is done using the K-Means iterative algorithm.
4. Centroid update: update the centroid in each cell using the centroid of the training vectors assigned to that cell.
5. Repeat steps 3 and 4 until the average falls below a preset threshold α.
6. Repeat steps 2, 3, 4 and 5 until a codebook of size M is reached.

4 Hidden Markov Model

A speaker verification system consists of two phases which is the training phase and the verification phase. In the training phase, the speaker voices are recorded and processed in order to generate its model for storage in the database. Meanwhile, in the verification phase, the existing reference templates are compared with the unknown voice input. In this project, we use the Hidden Markov Model (HMM) method as the training/recognition algorithm.

The most flexible and successful approach to speech recognition so far has been the HMM. The goal of HMM parameter estimation is to maximize the likelihood of the data under the given parameter setting. General theory of HMM has been given in [4,10,11,12,13]. There are 3 basic parameters in HMM which is:

- **π -** The initial state distribution.
- **a** – The state-transition probability matrix.
- **b** – Observation probability distribution.

In the training phase, an HMM for each speaker is generated. Each model is an optimized model for the word it represents. For example, a model for the word 'Satu' (number one), has its **a**, **b**, and **Π** parameters adjusted so as to give the highest probability score whenever the word 'Satu' is uttered, and lower scores for other words. Thus, to build a model for each speaker, a training set is needed. This training set consists of sequences of discrete symbols, such as the codebook indices obtained from the Vector Quantization stage.

Here, an example is given of how an HMM is used to build models for a given training set. Assuming that N speakers are to be verified, first we must have a training set of L token words, and an independent testing set. To do speaker verification, the following steps are needed:

1. First we build an HMM for each speaker. The L training set of tokens for each speaker will be used to find the optimum parameters for each word model. This is done using the re-estimation formula.
2. Then, for each unknown speaker in the testing set, first characterize the speech utterance into an observation sequence. This involves the use of an analysis method for the speech utterance so that we get some kind of feature vector, and then the vector is quantized using Vector Quantization. Thus, we will get a sequence of symbols, with each symbol representing the speech feature for every discrete time step.
3. We calculate **a**, **b** and **π** parameters for the observation sequence using one of the speaker models in the vocabulary. We then repeat for every speaker model in the database.

After N models have been created, the HMM engine is then ready for speaker verification. A test observation sequence from an unknown speech utterance (produced after vector quantization of cepstral coefficient vectors), will be evaluated using the Viterbi algorithm (the log-Viterbi algorithm is used to avoid precision underflow). For each speaker model, the probability score for the unknown observation sequence is computed. The speaker whose model produces the highest probability score and matches the ID claimed is then selected as the client speaker.

Speaker verification means making a decision on whether to accept or reject a speaker. To decide, a threshold T_i is used with each client speaker *i*. If the unknown speaker's maximum probability score exceeds this threshold, then the unknown speaker is verified to be the client speaker (i.e. speaker accepted). However, if the unknown speaker's maximum probability score is lower than this threshold, then the unknown speaker is rejected. The decision process is shown in Fig.1.

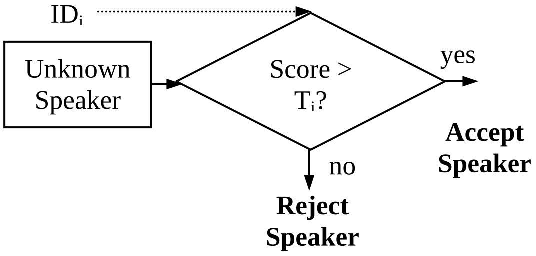


Fig.1 Speaker verification decision.

The threshold is determined as follows:

1. For each speaker, evaluate all samples spoken by him using his own HMMs and find the probability scores. From the scores, find the mean μ_1 and standard deviation σ_1 of the distribution.
2. For each speaker, evaluate all samples spoken by a large number of impostors (typically over 20) using the speaker's HMMs and finds the probability scores. From the scores, find the mean μ_2 and standard deviation σ_2 of the distribution.
3. For each speaker, the threshold is calculated as given below:

$$T_i = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} \qquad (2)$$

5 Experiments

Speaker verification experiments were carried out using the database described in section 3. As a comparison, speaker verification using the HMM approach and the hybrid HMM and VQ approach was carried out. To evaluate performance of the system in noisy environments, experiments using added Gaussian white noise at 4 levels (30dB, 20dB, 10dB and 0dB) were carried out.

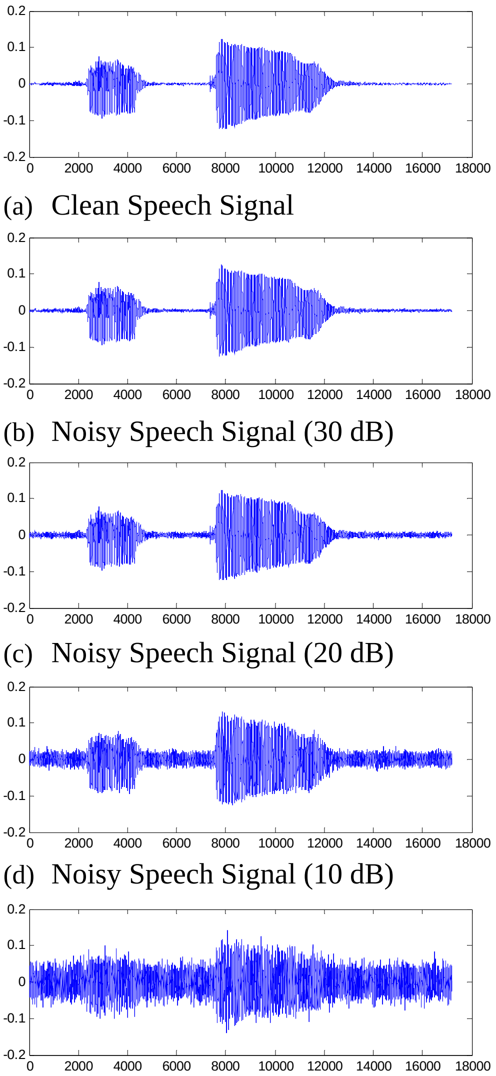
5.1 Experimental Conditions

For the experiments, 100 speakers were selected where each speaker has 10 repetitions of Malay digits. All of the Malay digits, from 0 until 9 were selected to build the speaker model. The samples were divided into 2 sets, one for the training session and the other for the testing session. During the first enrollment, 5 samples were selected to model the respective genuine speakers, yielding 100 different genuine models. In the second enrollment session, the remaining samples were used to generate the validation data in two different manners. In the testing session the validation data were used to derive a single genuine access by matching the utterance template with his own reference model, and use others to generate 99 impostor accesses. This simple strategy thus leads to 100 genuine and 9900 (100×99) impostors accesses, which are used to validate the performance of the individual verification system and to calculate the thresholds for the EER criterion.

Feature vectors composed of 14 linear predictive coding cepstral (LPCC) coefficients [7] were used. The 0^{th} coefficient was excluded, because it carries little speaker specific information. The analyzed frame was windowed by a 15 milliseconds Hamming window with 5 milliseconds overlapping. All samples were down-sampled to 16 kHz prior to feature extraction. The samples were pre-segmented automatically using the start-end detection module to remove the silent parts. For speaker modeling, all samples were selected from each speaker's training set. This procedure was for building the global codebook that will be used for HMM. Then, for each speaker, a codebook was built using the *Linde-Buzo-Gray* (LBG) VQ method. Therefore every speaker's codebook was built from 5 samples meant to represent the feature space occupied by each speaker as he utters unconstrained speech. The size of each codebook is 256 codevectors as for the global codebook. After the codebook for each speaker is built, the 5 speech samples were quantized to produce 5 symbol sequences. The symbol sequences were used to build an HMM for those particular genuine speakers. All models have 10 states, 256 symbols per state and the minimum symbol probability $b_j(0)$ is 0.00005. All models were re-optimized 5 times recursively. After the training process, there is a single digit model for each speaker. For testing we used a workstation, equipped with a Pentium D processor, with 1 GB of memory and running on the Windows XP operating system.

5.2 Experimental Conditions in Noisy Environments

Experiments in noisy environments were carried out using the same approach as in a clean environment (combination of VQ and HMM). Gaussian white noise was added to clean speech signals to produce noisy speech signals. Figure 2 (a) shows the clean signal of the digit 'Satu' (number one). Figure 2 (b) to (e) shows noisy speech signals mixed with Gaussian white noise with 30 dB to 0 dB SNR respectively.



(a) Clean Speech Signal

(b) Noisy Speech Signal (30 dB)

(c) Noisy Speech Signal (20 dB)

(d) Noisy Speech Signal (10 dB)

(e) Noisy Speech Signal (0 dB)

Fig. 2 Speech Signal of the word 'Satu' with different SNRs

6 Experimental Results and Discussions

6.1 Clean Environment

A total success rate (TSR) of 99.97% was achieved using this hybrid technique. For speaker verification, the true speaker rejection rate was 0.06% while the impostor acceptance rate was 0.03% and an equal error rate (EER) [14,15] of 11.72% was achieved. Table 2 shows a summary of the verification results for the experiments performed. Figure 3 shows an ROC plot of False Rejection Rate (FRR) vs False Acceptance Rate (FAR). It clearly shows that a hybrid technique of VQ and HMM outperformed the HMM based technique in all aspects.

Table 2 Verification result for clean environment (%)

SNR (dB)	Method	FRR	FAR	TSR	EER
Clean	HMM	25.30	9.99	89.87	16.66
	VQ+HMM	0.06	0.03	99.97	11.72

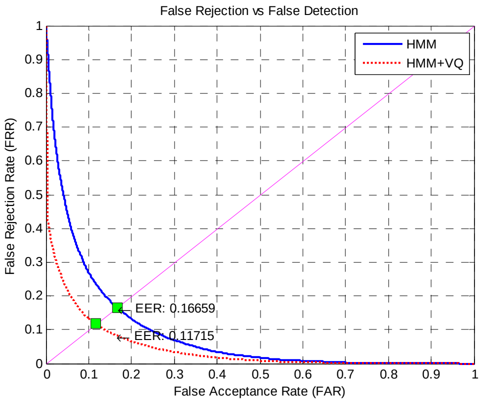


Fig.3 ROC plot of False Rejection Rate(FRR) vs False Acceptance Rate (FAR)

6.2 Noisy Environments

Table 3 Verification result of Gaussian white noise mixed (%)

SNR (dB)	Method	FRR	FAR	TSR	EER
0	HMM	34.47	57.15	43.07	49.94
	VQ+HMM	**59.02**	**37.23**	**62.57**	**49.11**
10	HMM	31.52	57.10	43.14	48.19
	VQ+HMM	**51.33**	**35.38**	**64.48**	**46.36**
20	HMM	28.12	55.42	44.83	45.21
	VQ+HMM	**40.78**	**28.99**	**70.90**	**42.35**
30	HMM	25.88	48.95	51.26	41.01
	VQ+HMM	**30.32**	**23.14**	**76.80**	**37.14**

Table 3 shows the verification result using the HMM and the hybrid HMM and VQ approaches in noisy environments (Gaussian white noise mixed). Using this combination approach, TSRs of 62.57, 64.48, 70.90 and 76.80 were achived for SNRs of 0 dB, 10 dB, 20 dB and 30 dB, respectively. High noise levels worsen the system performance in all cases. However, the hybrid technique of VQ and HMM outperformed the HMM based technique in all aspects.

7 Conclusion

The database used to test the speaker verification system has been described. The database was used in the testing and validation modules where experiments were performed in order to evaluate the system using an HMM and a hybrid VQ and HMM approaches in clean and noisy environments. It has been shown that in a clean environment, a total success rate (TSR) of 99.97% was achieved using this hybrid technique compared to an HMM which achieved 89.87% TSR. In noisy environments, TSRs of between 62.57%-76.80% were achieved for SNRs of 0-30 dB using the proposed technique compared to an HMM which was 43.07%-51.26%. As expected, in noisy environments both techniques showed degradation in performance compared to that in a clean environment. However, for both clean and noisy environments, the hybrid technique of VQ and HMM performed better when compared to an HMM.

8 Acknowledgement

This research is supported by the following research grant: Fundamental Research Grant Scheme, Ministry of Higher Education, FRGS UKM-KK-02-FRGS0036-2006

References:

[1] National Science and Technology Council (NTSC), “Speaker Recognition”, 2006. [Online]. http://www.biometricscatalog.org/NSTCSubcommittee/Documents/Speaker%20Recognition.pdf [7 August 2006].

[2] J.M. Naik, “Speaker Verification: A Tutorial”, *IEEE Communication Magazine*, January 1990, pp. 42-48.

[3] J.P. Campbell, “Speaker Recognition: A tutorial” , *Proc. of the IEEE*, Vol. 85, No. 9, September 1997, pp. 1437 – 1462.

[4] L.R. Rabiner and B.H. Juang, *Fundamental of Speech Recognition*, Prentice Hall, New Jersey, 1993.

[5] T. Matsui and S. Furui, “Comparison of Text Independent Speaker Recognition Methods using VQ-Distortion and Discrete/Continuous HMMs”, *Proceedings of ICASSP-92*, Vol. 2, 1992, pp. 157-160.

[6] M. Fujimoto and Y. Ariki, “Noisy Speech Recognition using Noise Reduction Method Based on Kalman Filter”, *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00 Proceedings. 2000 IEEE International Conference* **.** 2000, pp. 1727-1730.

[7] F.K. Soong, A.E. Rosenberg, L.R. Rabiner and B.H. Juang, “A Vector Quantization approach to Speaker Recognition”, Florida: ICASSP Vol.1, 1985, pp. 387-390.

[8] A. Vasuki and P.T. Vanathi, “A review of vector quantization techniques”, *Potentials IEEE*, July - August 2006, pp.39-47.

[9] Y. Linde, A, Buzo and R. M. Gray, “An algorithm for vector quantizer design”, *IEEE Trans. Commun.*, vol. COM-28, no.1, pp.84-95, Jan 1980.

[10] S. Yoshizawa, N. Wada, N. Hayasaka, Y. Miyanaga, “Scalable Architecure for Word HMM-BASED Speech Recognition”, *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on”* Volume 3, 23-26 May 2004 PP.III 417-420.

[11] W. Han et al. “An HMM-based speech recognition IC”, *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, Volume 2, 25-28 May 2003 PP. II 744-747 vol.2

[12] C. Wheddon and R. Linggard, *Speech and Language Processin*g, Chapman and Hall, UK, 1990, pp. 209-230.

[13] L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition”, *Proceeding of The IEEE*, Vol.77, No.2, February 1989.

[14] J.L. Wayman, “Error Rate Equations for the General Biometric System”. *IEEE Robotic & Automation.*, 6(9), March 1999, pp.35-48.

[15] M. Jin, F. K. Soong and C. D. Yoo, “A Syllable Lattice Approach to Speaker Verification*”, IEEE Transcation on Audio, Speech and Language Processing*, Vol.15, N0.8, November 2007, PP.2476-2484.