Journal of Quality and Technology Management
Volume XI, Issue I, June 2015, Page 97–106

COMPARISON OF SUB-MODEL CURTAILING TECHNIQUES TO ACCELERATE VECTOR QUANTIZATION BASED SPEAKER IDENTIFICATION

M. Afzal, T. Ahmad, M.F. Hayat, K.H. Asif, H.M. Shahzad
Department of Computer Science and Engineering,
University of Engineering and Technology, Lahore

ABSTRACT

Automatic speaker identification (ASI) is a remotely operative tacit technique for surveillance and tracking persons through digital telephone networks. Vector quantization (VQ) technique often performs in parity with Gaussian mixture model (GMM) in terms of accuracy and performs better in speed for automated speaker identification (ASI). Real-time speaker identification systems consume most of time comparing d-dimensional feature vectors extracted from a test speech sample with M codewords of codebooks of N registered speakers. Closest codeword search (CCS) is performed for N ×T times to find the best matching codeword for T number of feature vectors extracted from test speech sample to find the best matching registered speaker. It requires d-dimensional distance computations for M × N × T times. ASI speedup techniques focus on reducing the effect of parameters T, N, M or d. Vantage point tree (VPT) technique tends to reduce M by indexing codeword into a binary tree like structure to speedup CCS. Although best case speedup is expected to be $M\ /\log_2 M$ but best average speedup factor empirically found is reportedly only 1.67 for codebook size M=512. On the other hand partial distortion elimination (PDE) that had been mostly ignored in ASI focuses on reducing d. It has been observed that PDE reduces codebook size $M \times d$ by 3 times more than VPT to speedup speaker identification 3 times faster.

Keywords: *Speaker identification; vector quantization; distortion computation; vantage point tree.*

1) INTRODUCTION

Automatic speaker identification (ASI) systems find wide usage in credit card payment through internet, in security access control, personnel attendance where computer vision is not possible etc. Multi-user systems utilize an ASI front end to deliver better user specific services by adapting to the current user. These systems may not require exact identity of the speaker but identification of his or her class of speakers might be sufficient to make an intelligent decision for system adaptation (He X., 2000; Kuhn 2000). Automatic systems for speaker based segmentation of audio streams of legislative assembly proceedings, court room discussions and business meetings can use speaker identification subpart for this purpose. Surveillance and tracking of large number of wanted people for their appearance on the digital communication networks can be used to prevent crime and catch criminals through real-time ASI. Problem is that speaker identification time for a large number wanted persons on heavily loaded telephone networks is too large for real-time speaker identification. Thus real-time applications of ASI lay high emphasis on speeding up ASI. A number of techniques have been explored in (Kinnunen et. al., 2006) that tend to increase speed of speaker identification. These techniques manipulate speaker models as a whole. So, there is lack of research work on techniques that manipulate components with in speaker model. This paper discusses manifestation of ASI speedup techniques that operate on sub-components of speaker models. Such techniques that can also fully utilize the faster cache memory motivated this study.

1.1) Speaker Identification using VQ Technique

Major components of an ASI system based on VQ are shown in Figure 1. FE unit converts speech signal into a sequence feature vectors. These feature vectors are input to TR unit during training phase of the ASI system that trains codebooks of speakers to be registered with the system from their training speech samples. These codebooks which model registered speakers' speech are stored in the database DB. During testing phase FE unit converts test sample to feature vectors that are fed to PM unit. This switching function of routing output feature vectors to TR or PM unit is depicted by arc on the arrow head sticking out from FE unit. PM unit computes distortion of feature vectors of test sample with each codebook of registered speaker. The decision unit identify one of the registered speakers to be the test speaker whose codebook has minimum distortion with features of the test speech sample.

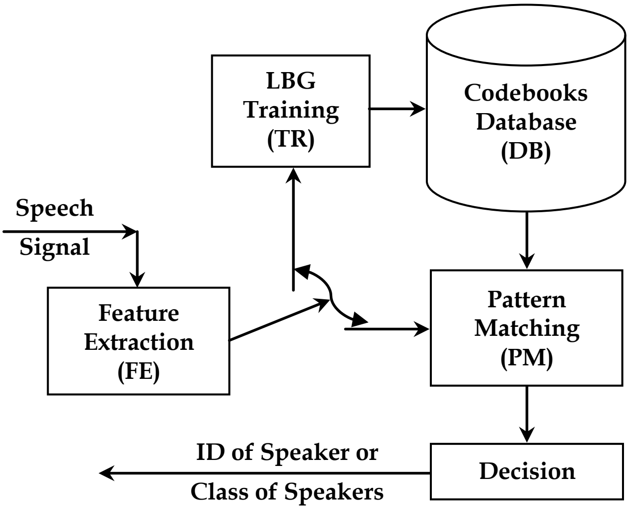

Figure 1: Major Components of an ASI System

Essentially, FE unit breaks speech sample into 70 to 100 frames per second with and overlap of 30% to 50% between consecutive frames. Mostly, Hamming window is applied to each frame before converting it to feature vector of up to 90 elements (Fan and Rosca, 2003). Mel-frequency cepstral coefficients (MFCC) are under frequent practice as feature vectors of size 10 to 20 (Kinnunen et al., 2006). While MFCC based feature vectors are extracted by getting magnitude discrete Fourier transform (DFT) of each windowed frame and filtering it through a triangular filter bank.

Output of the filter bank for each frame is log compressed before applying discrete cosine transform (DCT). First element of DCT representing energy is ignored and the remaining selected elements form a MFCC feature vector with dimension d of each vector. Let feature vectors extracted from speech samples of a speaker during training to be represented as $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_{\tilde{T}}) \mid \tilde{x}_i \in \mathbb{R}^d,\ \tilde{X} \in \mathbb{R}^{\tilde{T}\times d}$. Extracted feature vectors $\tilde{X}$ from speech samples of speaker s are trained by TM unit through clustering algorithm like Linde, Buzo and Gray (LBG) to quantize them to M mean vectors or codewords. Together set of these codewords form a codebook $C = (c_1, c_2, c_3, \dots, c_M) \mid c_i \in \mathbb{R}^d,\ C \in \mathbb{R}^{M\times d}$.

Let $X = (x_1, x_2, x_3, \dots, x_T) \mid x_i \in \mathbb{R}^d$ represent the sequence of T feature vectors extracted by FE unit from speech samples of an unknown (test) speaker. Similarity measure of X with codebook C of a registered speaker is computed in terms of average quantization distortion using equation (1) as reported (Quatieri, 2002; Kinnunen et. al., 2006).

$$D_{avg}\ (X, C) = \frac{1}{T}\sum_{i=1}^{T} e(x_i, C) \ \mid e(x_i, C) = \min_{c_m \in C} \Delta(x_i, c_m) \qquad (1)$$

Where $e(x_i, C)$ represents distortion of a test vector, x_i with a codebook C of target registered speaker model. Here, $\Delta(x_i, c_m)$ stands for distance between the test vector, $x_i \mid 1 \le i \le T$ and a codeword, $c_m \mid 1 \le m \le M$ of codebook C. The test speaker is identified as one of the registered speaker whose model has the minimum average distortion. Various other similarity measures have also been suggested by (Fan and Rosca, 2003; Matsui and Furui, 1991; Wang et al., 1990).

The vector distortion $e(x_i, C) = \min_{c_m \in C} \Delta(x_i, c_m)$ for a test vector $x_i \in \mathbb{R}^d$ and codebook $C \in \mathbb{R}^{M \times d}$ is computed using Euclidean distance as given by Equation (2).

$$\Delta(x_i, c_m) = \|x_i - c_m\| \qquad (2)$$

Finding the Average Distortion of a test vector sequence $X = (x_1, x_2, x_3, ..., x_T)$ with codebook $C = \{c_1, c_2, c_3, ..., c_M\}$ of a registered speaker requires $M \times T$ computations of d-dimensional distance. In case of full search procedure, distortion computation with codebooks of N registered speakers involves $N \times M \times T$ d-dimensional distance computations. Subsequently minimum average distortion is computed. Thus the total computation involves $2 \times d \times N \times M \times T$ additions and $d \times N \times M \times T$ multiplications giving time order of computational complexity as $O(d \times N \times M \times T)$.

1.2) Speeding up VQ Based ASI Systems

Each Euclidean distance computation for d-dimensional test vectors requires d multiplications and $2 \times d$ additions. To find distortion $e(x_i, C)$

between a single test vector x_i and a codebook C requires M distance computations. Time order complexity of $e(x_i, C)$ computation can be given as $O(d \times M)$. Idea behind speeding up techniques, focusing on reducing time for vector distortion computations studied for ASI systems follow two directions as described next.

Vantage Point Tree (VPT) Indexing

Kinnunen et al., (2006) have studied Vantage Point Tree (VPT) indexing technique to speedup the vector distortion $e(x_i, C)$ computation. The VPT indexing technique stores codeword of a codebook in an indexed tree structure so that average number of *d*-dimensional Euclidean distance computation is reduced to $\log_2(M)$. Using VPT, $e(x_i, C)$ computation is expected to have time order as $O(d \times \log_2 M)$ for a fully balanced binary tree structure.

Experimental results of Kinnunen et al., (2006) to speedup speaker identification on TIMIT (Garofolo et al., 1993) speech corpora show that time reduction does not follow logarithmic order for computing $e(x_i, C)$ hence effective time order is rather given as $O(d \times M \times \eta_{VPT})$ where η_{VPT} is codebook reduction factor of VPT indexing scheme. Explanation of TIMIT speech corpora is made in **speech material** section.

Partial Distortion Elimination (PDE)

Partial distortion computation algorithm (Bei and Gray, 1985) computes distance of the vector x_i with first codeword c_1 of the codebook and sets it as the threshold distortion e'. Next *M*-1 distances are computed incrementally checking $\forall 1 \le j \le d$ the accumulating sum $s = \sum_{j=1}^{d} (x_{i,j} - c_{m,j})^2$. If $s \ge e'$ distance computed is discarded and distance computation for the next codeword is started. If $s < e', j = d$ the threshold is updated $e' = s$ and the process is repeated till *m*=*M* to finalize vector distortion $e(x_i, C) = e'$. Effectively time order of computing $e(x_i, C)$ becomes $O(d \times M \times \eta_{PDE})$, where η_{PDE} is the codebook reduction factor for PDE algorithm. PDE was first time used for speeding up ASI (Afzal and

Haq, 2010). PDE is very simple to implement and highly efficient as compared to VPT. This paper compares VPT and PDE to demonstrate how two techniques differ in pruning the sub-components of VQ speaker models called codebooks. The main objective is to lay emphasis on PDE, so that further research can be motivated regarding efficient cache usage and parallel implementations on massively parallel computer systems.

2) METHODOLOGY

2.1) Speech Material

We acquired TIMIT speech data from Linguistic Data Consortium (LDC), Pennsylvania University, USA for empirical study of achievable speed up by PDE in this paper. TIMIT data consists of read speech samples of English language from 630 speakers consisting of 192 female and 438 male speakers recorded with microphone. There are 10 speech sample files of each TIMIT speaker. Two 'sa' and five 'sx' of TIMIT files that contained same phonetic contents for each speaker were used for system training to build VQ codebooks. Three 'si' TIMIT files that have different phonetic contents for each speaker were used to test our trained ASI systems. It may further explain the TIMIT data set that 'sa', 'sx' and 'si' are file names for each TIMIT speaker's sample. This files selection allowed us to conduct our experiments for speaker identification in text independent mode. Average duration of speech sample for system training and testing was 22.4 seconds and 8.4 seconds respectively.

2.2) Feature Extraction, Model Training and Pattern Matching

We down sampled TIMIT speech corpus to 8 kHz through anti-aliasing filter in our experimentation. Digital speech samples broken into frames of 30 milliseconds duration had 33% overlap between consecutive frames. Energy of each frame of digital speech signal was computed. Speech signal frames with energy less than 1.5% of the average frame energy were discarded as silence. This threshold frame energy based silence removal criteria reduced total number of speech frames extracted from training and testing samples by 10% and 8.5% respectively. Discrete Fourier Transform (DFT) of each non-silence frame was taken after applying Hamming window. Approximation of Mel-frequency scale distribution along the frame frequency spectrum defined by Equation (3) was used to make 19 triangular filters.

$$f_{Mel} = 2595\ \log_{10}(1 + f_{Lin} / 700) \tag{3}$$

Subsequently outputs of Mel-frequency triangular filter banks was log compressed and DCT was taken. The first value was ignored and next 12 values of DCT cepstrum were selected as 12-dimensional MFCC feature vectors. This dimension of MFCC vectors was selected to facilitate comparison with results reported in (Kinnunen et al., 2006). MFCC vectors extracted from test samples were also stored for use in different test runs. LBG algorithm was used to compute and store VQ codebook models of the speakers from feature vectors extracted from training samples.

2.3) Performance Testing

Silence removal threshold energy factor of 1.5% and 19 size filter bank were set after extensive training and testing for VQ model of size 64 for maximum accuracy. Identification accuracy was measured by identification error rate.

All programs for feature extraction, LBG algorithm and speaker identification with minimum distortion were made using Microsoft Visual C#. Hardware used for this purpose was HP Compact dx7400 Micro tower with Intel(R) Core(TM)2 Duo CPU E6550 @2.33 GHz with 2.00 GB memory installed. Windows Vista Business 32-bit version (2007), Service Pack 1 that installed on the HP machine has been used. Identification time for all 630 TIMIT speakers was computed by calling 'DateTime.Now' function of Microdsoft.NET framework library.

3) RESULTS AND DISCUSSIONS

Results of tests of different experiments performed on TIMIT speech data for close-set speaker identification are shown in Table 1 and Table 2 for PDE and VPT respectively.

Table 1: Accuracy and average speedup factors for VQ codebooks with PDE on TIMIT data

Codebook Size ($M{\times}d$)	Error Rate %	Times Seconds Baseline	Times Seconds PDE	Speedup Factor	Effective Curtailed Model Size M' or d'	
					M'	d'
32x12	14.92	0.74	0.26	2.90:1	11.03	4.14
64x12	4.92	1.43	0.45	3.17:1	20.19	3.79
128 x12	1.27	2.79	0.81	3.45:1	37.01	3.47
256 x12	0.32	5.50	1.47	3.73:1	68.63	3.22
512 x12	0.48	11.00	2.81	3.92:1	130.61	3.06

Table 2: ASI Performance of Vantage Point Tree (VPT) Experiments on TIMIT Database

Model Size ($M{\times}d$)	Error Rate %	Time Seconds Baseline	Time Seconds VPT	Speedup Factor	Effective Model Size M'
32 x12	14.92	0.74	0.72	1.04:1	30.77
64 x12	4.92	1.43	1.32	1.09:1	58.72
128 x12	1.27	2.79	2.41	1.16:1	110.34
256 x12	0.32	5.50	4.40	1.24:1	206.45
512 x12	0.48	11.00	6.47	1.70:1	301.18

Identification accuracy increases with model size but decrease for codebook of size 512 due to over fitting in both PDE and VPT cases. Such behaviour is also observed in (Kinnunen et al., 2006). PDE speedup factor increases monotonously with increase in model size. VPT speedup factor also increases monotonously with increase in model size. Our VPT speedup factors are almost the same as those shown in (Kinnunen et al., 2006). Anyhow speedup factors of PDE are much larger than those due to VPT. Speeding up due to PDE has been translated into reduced codebook size (**d'** and **M')** values to calculate effective model size due to computation curtailing so that PDE and VPT can be compared on a common criteria.

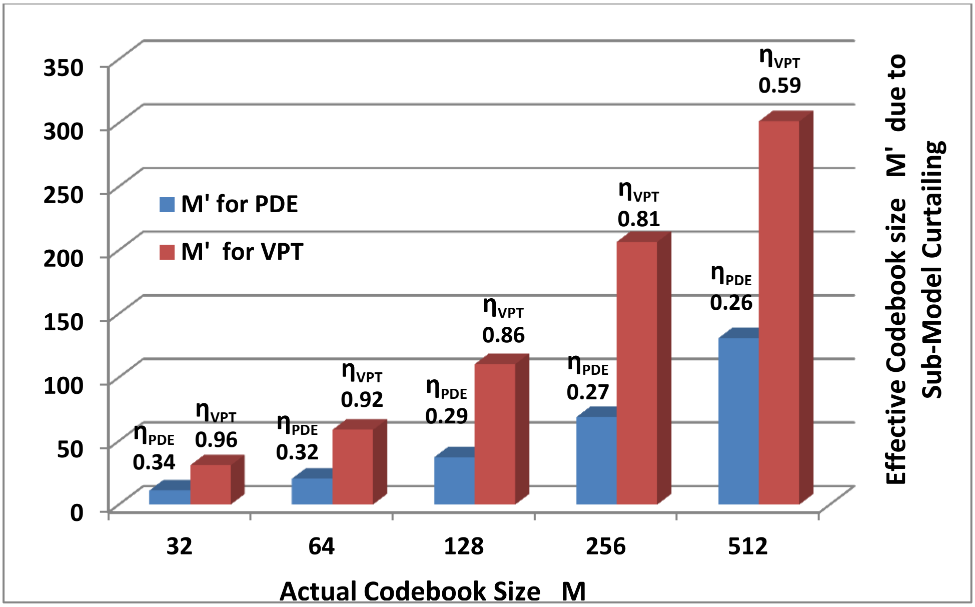

Figure 2: Comparison of Codebook Curtailing achieved by PDE and VPT

Figure 2 gives a pictorial view of comparison between VPT based reduction in codebook size as η_{VPT} and PDE based reduction in codebook size as η_{PDE} with respective to different actual codebook sizes.

4) CONCLUSIONS

This paper presented a comparative analysis of capabilities of VPT and PDE algorithms to speedup speaker identification process based on vector quantization. Our analysis showed that PDE is 3 times more effective than VPT in speeding up ASI. It is, therefore, recommended that PDE should be studied in combination with various speaker model curtailing techniques to further speedup speaker identification process. It can be concluded that PDE is very simple to implement and it can be applied to substantially curtail speaker model which results in speeding up VQ based speaker identification for smaller as well as large sized speaker models. Also that PDE is equally good for smaller as well as larger codebooks while VPT offers diminishing returns when applied to smaller codebooks.

REFERENCES

Afzal, M. and Haq S., 2010. Accelerating Vector Quantization Based Speaker Identification, Journal of American Science, 6(11) pp.1046-1050.

Bei C. and Gray R., 1985. An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization. IEEE Transactions on Communication. (33) 10, pp.1132-1133.

Fan N., and Rosca J., 2003. Enhanced VQ-Based Algorithms For Speech Independent Speaker Identification. In: Proc. Audio- and Video-Based Biometric Authentication, pp. 470–477.

Garofolo J., Lamely L., Fisher W., Fescues J., Pallet D., Dahlgren N., and Zoë V.,1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus.

He X., and Zhao X., 2000. Fast Model Selection Based Speaker Adaptation for Non-native Speech. IEEE Transactions on Speech Audio Process, 11 (4), pp.298-307.

Kinnunen T., Karpove E., and Franti P., 2006. Real-Time Speaker Identification and Verification. IEEE Transactions on Audio and Language Processing, 14 (1), pp.277-288.

Kuhn R, J, Junqa C, Nguyen P, and Niedzielski N. 2000. Rapid Speaker Adaptation in Eigen voice Space. IEEE Transactions on Speech Audio Process, 8 (9), pp.695-707.

Linde Y., Buzo A., and Gray R., 1980. An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications, 28 (1), pp.84–95.

Matsui T., and Fruit S., 1991. A Text-Independent Speaker Recognition Method Robust Against Utterance Variations. Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, pp.377–380.

Quatieri T., 2002. Discrete-time Speech Signal Processing Principles and Practice., Pearson Education Inc.

Wang R., He L., and Fujisaki H., 1990. A Weighted Distance Measure Based On the Fine Structure of Feature Space: Application to Speaker Recognition. In: Proc. Int. Conf. on Acoustics, Speech and Signal Processing, pp.273–276.