Accelerating Vector Quantization Based Speaker Identification

Muhammad Afzal[1], Shaiq A. Haq[2]

[1]Department of Computer Science and Engineering,
University of Engineering and Technology, Lahore-54890, Pakistan

[2]Dean Faculty of Engineering, Wah Engineering College,
University of Wah, Wah Cantt., Pakistan

E-mails: shmafzal@yahoo.com, shaiq_haq@yahoo.com

Abstract: Matching of feature vectors extracted from speech sample of an unknown speaker, with models of registered speakers is the most time consuming component of real-time speaker identification systems. Time controlling parameters are size and count of extracted test feature vectors as well as size, complexity and count of models of registered speakers. We studied vector quantization (VQ) for accelerating the bottlenecking component of speaker identification which is less investigated than Gaussian mixture model (GMM). Already reported acceleration techniques in VQ approach reduce test feature vector count by pre-quantization and reduce candidate registered speakers by pruning unlikely ones, thereby, introducing risk of accuracy degradation. The speedup technique used in this paper partially prunes VQ codebook mean vectors using partial distortion elimination (PDE). Acceleration factor of up to 3.29 on 630 registered speakers of TIMIT 8kHz speech data and 4 on 91 registered speakers of CSLU speech data is achieved respectively.
[Muhammad Afzal, Shaiq A. Haq. Accelerating Vector Quantization Based Speaker Identification, Journal of American Science 2010;6(11):1046-1050]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Speaker identification, vector quantization, partial distortion elimination, speaker pruning.

1. Introduction

Automated Speaker Identification (ASI) systems identify a test speaker from the database of its registered speakers (Quatieri, 2002). ASI systems have three major units namely Feature Extraction (FE), Model Training (MT) and pattern matching (PM) as shown in Figure 1.

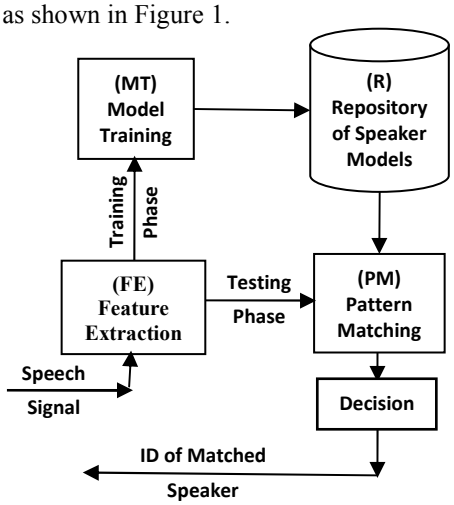

Figure 1. Major Components of an ASI System

FE unit is used both by MT and PM units as front processor. The input to FE unit is a digital speech signal which is converted by it to a sequence of *d*-dimensional vectors each consisting of *d* values of speaker specific features. Mostly Mel-frequency Cepstral Coefficients (MFCC) feature vectors of 12 to 20 elements are used (Kinnunen, 2006). MT unit of VQ based ASI systems compresses feature vector sequence $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, ..., \tilde{x}_{\tilde{T}})$ of size $\tilde{T}$ to smaller number of mean vectors by generally implementing Linde Buzo Gray (LBG) clustering algorithm (Bei and Gray, 1985). The set of *M* mean vectors is termed as codebook, $C \in \mathbb{R}^{M \times d}$. For an ASI system of *N* registered speakers, *N* codebooks are computed and stored in a repository, R, mathematically given by Expression (1).

$$R \underset{store}{\leftarrow} \left\{ \sum_{LBG}^{N} \tilde{X} \Rightarrow C \right\} \qquad (1)$$

Where $\mathbb{R}$ represents real number space, $\tilde{X} \in \mathbb{R}^{\tilde{T} \times d}$, $R \in \mathbb{R}^{N \times M \times d}$ and; *d, M* and *N* are as defined above.

VQ codebook is called non-parametric model while GMM is termed as parametric model. GMM training is mostly initialized with LGB clusters

to determine its parameters using expectation maximization (EM) algorithm (Alpaydin, 2004). GMM based speaker recognition systems have been extensively studied for improving speed (Kinnunen et al., 2006). In this paper we present speeding results for VQ based systems which are as efficient as GMM (Kinnunen and Li. 2009).

Full search based PM unit of VQ system computes Δ, d-dimensional Euclidean distances between each vector of sequence of the test feature vectors, $X = (x_1, x_2, x_3, ..., x_T)$, and each of the mean vector of each target registered speaker's codebook stored in repository, R, using Equation (2). Where T is the number of feature vectors extracted from the samples and $X \in \mathbb{R}^{T \times d}$. Euclidean distances are used to compute similarity measure called single vector distortion $D_{t,s}$ between each test vector x_t and each stored target codebook of speaker s, R_s, as given by Equation (3). Identification decision is done using Equation (4).

$$\forall \begin{cases} 1 \le t \le T, \\ 1 \le s \le N, \\ 1 \le m \le M \end{cases} \quad \Delta_{t,s,m} = \sqrt{\sum_{i=1}^{d} (X_{t,i} - R_{s,m,i})^2} \qquad (2)$$

$$\forall 1 \le t \le T, 1 \le s \le N \quad D_{t,s} = \arg\min_{1 \le m \le M} \Delta_{t,s,m} \qquad (3)$$

$$Decision \quad Speake.r \quad id = \arg\min_{1 \le s \le N} \sum_{t=1}^{T} D_{t,s} \qquad (4)$$

Full search speaker identification as given by Equations (2)-(4) shows that $T \times N \times M \times d$ multiplications, $2 \times T \times N \times M \times d$ additions and $T \times N \times M$ square root computations are required. Identification time order can be given by $O(T \times N \times M \times d)$. Such high time order complexity of minimum distortion slows down the identification process. Real-time speech processing systems require fast speaker identification front-end to adapt to speaker specific speech model. This emphasizes the need for research to accelerate speaker recognition task.

Brief review of existing accelerating techniques for ASI systems is given in section 2. Algorithm used to speedup ASI system that partially prunes codebooks along with its performance analysis is presented in section 3. Description of speech material used in this study, experimental setup and its parameters are given in section 4. Results of experimental are shown and discussed in section 5 followed by conclusions in section 6.

2. Existing Techniques

Inserting $\Delta_{t,s,m}$ definition for EUD from Equation (2) into Equation (3) reduces square root computations from $T \times N \times M$ to $T \times N$ as shown by Equation (5)

$$D_{t,s} = \sqrt{\arg\min_{1 \le m \le M} \sum_{i=1}^{d} (X_{t,i} - R_{s,m,i})^2} \qquad (5)$$

Reducing T by silence detection in raw speech signal is a normal practice. Further, best speedup techniques as reported by Kinnunen et al (2006) reduce T by pre-quantization (PreQ) of test vector sequence. They have used Vantage Point Tree (VPT) indexing technique to avoid mean vectors of codebooks in searching closest of M mean vectors. They used probabilistic measure to reduce N by pruning unlikely speakers.

Table 1. Parameters and Results of Kinnunen et al. (2006) Experiments on TIMIT Database

Code Book Size	Speedup Technique	Error Rate %	Times (S)	Speedup Factor
32	**Baseline**	**0.63**	**1.15**	**1 : 1**
	VPT+PreQ	0.63	1.11	1.04 : 1
	VPT+ Pruning	- -	- -	- -
64	**Baseline**	**0.48**	**2.37**	**1 : 1**
	VPT+PreQ	0.64	0.48	4.9 : 1
	VPT+ Pruning	**0.48**	**0.43**	**5.5 : 1**
128	**Baseline**	**0.16**	**4.82**	**1 : 1**
	VPT+PreQ	0.64	0.59	8.2 : 1
	VPT+ Pruning	**0.00**	**1.88**	**2.6 : 1**
256	**Baseline**	**0.16**	**10.2**	**1 : 1**
	VPT+PreQ	0.64	1.18	8.6 : 1
	VPT+ Pruning	**0.00**	**3.28**	**3.1 : 1**

Information specific to the test speaker is distributed all along the test vector sequence and pre-quantization of test vectors is likely to distort it as shown in Table 1 by test results by (Kinnunen, et al., 2006) for VPT+PreQ.

Table 1 shows absolute identification time and speedup ratio in (Kinnunen, et al., 2006) for

different speedup techniques exercised on a cluster of 2 Dell Optiplex G270 computers having 2.8 GHz processor and 1 GB RAM each. In Table 1 the effect of Vantage Point Tree (VPT) for speedup is multiplied with their other speedup algorithmic steps to simplify comparison with our results. We use PDE to speedup ASI system rather than VPT and speaker pruning as a whole.

3. Speedup Technique Used

Let 'SI' stand for identity number, ***id***, of the best matching registered speaker, more specifically, the candidate speaker, and **'Dmin'** stand for the minimum distortion of the candidate speaker. Algorithm presented next speeds up computation for Equation (5) by avoiding mathematical operations when ever possible and outputs the **id** of the test speaker.

Neighborhood search for closest mean vector to a feature vector, as expressed by Equation (3), is made faster by PDE algorithm proposed by Bei and Gray (1985). PDE algorithm has been largely employed in image compression for encoding and decoding images (Lee and Chen. 1994). We investigated its capability for speeding up speaker identification in partially pruning mean vectors that are unlikely to be nearest neighbor of a test feature vector, x_t, in the process of computing $D_{t,s}$ for modeling of any speaker *s*. Effectively, PDE reduces parameter *d* in time order complexity $O(T \times N \times M \times d)$.

Embedded PDE in the presented algorithm avoids superfluous multiplications and twice as many additions, whenever `D2≥D2m` causes Prune Events (PE1) or (PE2) by termination of EUD computation for current value of *m* and initiation of distance computation for (*m*+1). Line labels PE1 and PE2 used in the algorithm correspond to prune events that occur during the algorithm execution. In hypothetically best case $(M-1)(d-1)$ multiplications are avoided if PE1 or PE2 occur at *i*=1 for $\forall\ 2 \le m \le M$. In the worst case no multiplication or addition is avoided if PE1 or PE2 never occurs. In general $D_{t,s}$ is computed with partial scan through the speaker model. It follows from best and worst cases that average case speedup factor, given by $\frac{2 \times M \times d}{M \times d + M + d - 1}$, is less than 2.

The following algorithm outputs ***id*** of test speaker and requires input of test feature vectors, $X \in \mathbb{R}^{T \times d}$, and repository of codebooks of registered speakers, $R \in \mathbb{R}^{N \times M \times d}$. Square brackets are used for indices rather than subscripts.

Algorithm: VQ ASI with embedded PDE

```
SI ←1
Dmin ← 0
for t←1 to T do  D2m←0;
  for i ← 1 to d do
    dif ← X[t][i]-R[1][1][i]
    D2m ← dif×dif + D2m
  endfor
  for m ← 2 to  M  do
    D2 ← 0;
    for i ← 1  to d  do
      dif ← X[t][i]-R[1][m][i]
      D2 ← dif×dif + D2
      if D2 ≥ D2m goto PE1
    endfor
PE1: if D2 < D2m then D2m ← D2
  endfor
  Dmin ← sqrt(D2m) + Dmin
endfor
for s ←2 to N  do
  Dsum ← 0
  for t ← 1 to T  do
    D2m←0;
    for i ← 1 to d  do
      dif ← X[t][i]-R[s][1][i]
      D2m ← dif×dif + D2m
    endfor
    for m ← 2 to  M  do
      D2 ← 0;
      for i ← 1  to d  do
        dif ← X[t][i]-R[s][m][i]
        D2 ← dif×dif + D2
        if D2 ≥ D2m goto PE2
      endfor
PE2:  if D2 < D2m then D2m ← D2
    endfor
    Dsum ← sqrt(D2m) + Dsum
  endfor
     if Dsum<Dmin
     then
        Dmin ← Dsum;
        SI ←s
     endif
endfor
OUTPUT(SI);
```

4. Experiment

TIMIT (Garofolo et al., 1993) speech data was down sampled to 8kHz using anti-aliasing filter to match with sampling frequency of CSLU (Cole et al., 1998) data. Three TIMIT 'si' files were concatenated to get 8.4 seconds long test sample on the average. TIMIT data consists of read speech of microphone recordings. Hence speaker recognition results for TIMIT data are highly optimistic. For the purpose of validation we used CSLU speaker recognition corpus that consisted of telephonic speech in response to prompts. In total 40 prompts, labeled by two letters e.g., 'aa', 'aq' etc., were sent to participant speakers and their response speeches mostly repeated 4 times by the speakers were recorded over telephone. Some prompts were not sent to all the participants for unknown reasons in each of 12 sessions distributed over two year interval. Speech data of first four sessions was used in our experiments. For testing speech data files with prompts labeled as 'aa', 'ab', 'ac', 'am', 'an', 'ao' and 'av' were selected from sessions 2, 3 and 4. Average duration of speech per speaker was 28 seconds.

For system training all 'sa' and 'sx' TIMIT files were concatenated to get approximately 23 second long speech samples. While from CSLU corpus files corresponding to prompts labeled as 'aq', 'ar', 'as', 'at', 'au', 'be', 'bf', 'bg', 'bh' and 'bi' from sessions 1-4 were used. Total average duration of speech data per speaker was 99 seconds. Speech data selection thus made, allowed all the experiments for speaker identification to be conducted in text independent mode.

MFCC feature vector extraction was done by standard process (Deller et al., 2000). Hamming window was applied on 33% overlapping frames. Energy based silence detection was used for all tests. Raw speech frames were reduced by 9% and 8% from training and testing samples respectively for TIMIT while for CSLU data the values were 8.4% and 4.8% respectively.

A bank of 19 triangular filters was applied on magnitude real DFT spectrograms of 30 millisecond speech frames. MFCC vectors of size d=12 were computed from response of triangular filterbank once and stored for use both in training and testing for TIMIT. For CSLU speech data that had undergone telephone degradation, first 3 and last 2 triangular filters were not applied. Consequently frequencies between approximately 230 Hz to 3185 Hz were processed.

VQ codebook repository was prepared using LBG algorithm for all 630 TIMIT and 91 CSLU speakers from MFCC feature vectors extracted from training data. LBG trained codebooks with M = 32, 64, 128, 256, 512 were computed once and stored to use in testing. For CSLU data, codebooks with M=1024, 2048 were also trained. All algorithms were coded in Microsoft C#. Programs were run on 32-bit Windows Vista(TM), installed on HP Compac DX7400 with Intel(R) Core(TM)2 Duo CPU E6550 @2.33 GHz with 2 GB RAM. Time intervals were computed by calling 'System.DateTime.Now' method of C#.

5. Results and Discussion

Test results for speaker identification for TIMIT and CSLU corpora are shown in Table 2 and Table 3 respectively.

Table 2: Average Speaker Identification Performance for TIMIT data

VQ System		TIMIT DATA		
Model Size	Search Type	Error %	Time (S)	Speedup Factor
32	Baseline	15.71	1.25	1 : 1
	PDE	14.92	0.52	2.40 : 1
64	Baseline	5.40	2.45	1 : 1
	PDE	4.92	0.95	2.58 : 1
128	Baseline	1.27	4.84	1 : 1
	PDE	1.27	1.74	2.78 : 1
256	Baseline	0.32	9.61	1 : 1
	PDE	0.32	3.17	3.03 : 1
512	Baseline	0.48	19.15	1 : 1
	PDE	0.48	5.83	3.29 : 1

Table 3: Average Speaker Identification Performance for CSLU data

VQ System		CSLU DATA		
Model Size	Search Type	Error %	Time (S)	Speedup Factor
32	Baseline	6.59	0.31	1:1
	PDE	6.59	0.12	2.58:1
64	Baseline	2.20	0.64	1:1
	PDE	2.20	0.25	2.56:1
128	Baseline	0.00	1.13	1:1
	PDE	0.00	0.38	2.97:1
256	Baseline	0.00	2.21	1:1
	PDE	0.00	0.69	3.20:1
512	Baseline	0.00	4.35	1:1
	PDE	0.00	1.27	3.43:1
1024	Baseline	0.00	9.69	1:1
	PDE	0.00	2.73	3.55:1
2048	Baseline	0.00	19.36	1:1
	PDE	0.00	4.95	3.91:1

VQ models larger than 512 for TIMIT are not made since count of feature vectors extracted from training sample is less than 1024.

Accuracy of speaker identification increases with codebook size from 32 to 256. Systems, with codebook size 512 of TIMIT data, show over fitting degradation effects, as reported by Kinnunen et al (2006). Test results of our speedup technique with PDE show that it did not degrade accuracy when compared with corresponding full search (Baseline) systems. The technique is applicable on larger as well as smaller models. Speedup factor increases with increase in model size.

Identification accuracy for CSLU data is higher than corresponding TIMIT data that may be due to less number of speakers in CSLU data than that in TIMIT data. Speedup factor of PDE increases monotonously with codebook size for both TIMIT and CSLU data. Whereas in (Kinnunen, et al., 2006) speedup factor decreases from model size 64 to 128 and than increases for 256. In case of CSLU data there is no over fitting accuracy degradation for larger codebooks. PDE speedup factors of our systems corresponding to VPT+Pruning speedup factors shown in (Kinnunen, et al., 2006) are better in general. It is noteworthy that experimentally achieved average speedup factors of FDE for all codebook sizes are greater than theoretically possible factor 2.

6. Conclusions

Performance of a simple to implement technique, PDE, as compared to VPT and speaker pruning techniques given in (Kinnunen, et al., 2006), for speeding up VQ based real-time speaker identification systems, is presented in this paper. PDE is used to partially prune speaker models by obviating full scan of mean vectors of codebooks. Overall speedup factor of up to 4 is achieved. The time order, $O(T \times N \times M \times d)$ parameter d that is ignored in (Kinnunen, et al., 2006) can be successfully manipulated to speedup ASI systems. PDE can be applied to substantially speedup VQ based speaker identification for small as well as large sized models.

Acknowledgements:
This research was fully supported by the University of Engineering and Technology, Lahore, Pakistan. Their support is gratefully acknowledged.

TIMIT data was provided by Linguistic Data Consortium, University of Pennsylvania, USA. Their support is also gratefully acknowledged.

Corresponding Author:
Muhammad Afzal
Department of Computer Science and Engineering,
University of Engineering and Technology,
Lahore-54890, Pakistan
E-mail: shmafzal@yahoo.com

References

1. Alpaydin E. Introduction to Machine Learning. MIT Press, 2004.

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

3. Cole R, Noel M, Noel V, The CSLU Speaker Recognition Corpus. Proc. 5th Int. Conference on Spoken Language Processing (ICSLP), Sydney, Australia, (1998) pp.3167-3170.

4. Deller J R. Hansen H L. Proakis J G. Discrete-Time Processing of Speech Signals. IEEE Press, New York, 2000.

5. Garofolo J S, Lamel L F, et al.: TIMIT Acoustic-Phonetic Continuous Speech Corpus. 1993 http://www.ldc.upenn.edu/

6. Kinnunen T, Li H. An Overview of Text-Independent Speaker Recognition: from Features to Supervectors. Speech Communication. Elsevier, July, 2009.

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

8 Lee C-H., and Chen L-H., 1994. Fast Closest Codeword Search Algorithm For Vector Quantization. IEE Proc.-Vis. Image Signal Processing 141 (3), pp.143-148.

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

10/5/2010