Scalable Distributed Speech Recognition Using Multi-Frame GMM-Based Block Quantization

Kuldip K. Paliwal and Stephen So

School of Microelectronic Engineering,
Griffith University, Brisbane, Australia, 4111.
k.paliwal@griffith.edu.au
s.so@griffith.edu.au

Abstract

In this paper, we propose the use of the multi-frame Gaussian mixture model-based block quantizer for the coding of Mel frequency-warped cepstral coefficient (MFCC) features in distributed speech recognition (DSR) applications. This coding scheme exploits intraframe correlation via the Karhunen-Loéve transform (KLT) and interframe correlation via the joint processing of adjacent frames together with the computational simplicity of scalar quantization. The proposed coder is bit-rate scalable, which means that the bit-rate can be adjusted without the need for re-training of the quantizers. Static parameters such as the probability density function (PDF) model and KLT orthogonal matrices are stored at the encoder and decoder and bit allocations are calculated 'on-the-fly' without intensive processing. This coding scheme is evaluated in this paper on the Aurora-2 database in a DSR framework. It is shown that this coding scheme achieves high recognition performance at lower bit-rates, with a word error rate (WER) of 2.5% at 800 bps, which is less than 1% degradation from the baseline word recognition accuracy, and graceful degradation down to a WER of 7% at 300 bps.

1. Introduction

With the increase in popularity of remote and wireless devices such as personal digital assistants (PDAs) and cellular phones, there has been a growing interest in applying automatic speech recognition (ASR) technology in the context of mobile communication systems. Speech recognition can facilitate consumers in performing common tasks, which have traditionally been accomplished via buttons or pointing devices, such as making a call through voice dialing or entering data into their PDAs via spoken commands and sentences. Some of the issues that arise when implementing ASR on mobile devices include: computational and memory constraints of the mobile device; network bandwidth utilization; and robustness to noisy operating conditions.

Mobile devices generally have limited storage and processing ability which makes implementing a full on-board ASR system impractical. The solution to this problem is to perform the complex speech recognition task on a remote server that is accessible via the network. Various modes of this client-server approach have been proposed and reported in the literature. In the *Network Speech Recognition* (NSR) mode [1], the user's speech is compressed using conventional speech coders (such as the GSM speech coder) and transmitted to the server which performs the recognition task. In speech-based NSR (Fig. 1(a)), the server calculates ASR features from the decoded speech to perform the recognition. In bitstream-based NSR (Fig. 1(b)), the server uses ASR features that are derived from linear predictive coding (LPC) parameters taken directly from the bitstream. Numerous studies have been reported in the literature evaluating and comparing the performance of these two forms of NSR [2, 3, 4, 5, 6, 7, 8].

In *Distributed Speech Recognition* (DSR), shown in Fig. 1(c), the ASR system is distributed between the client and server. Here, the feature extraction of speech is performed at the client. These ASR features are compressed and transmitted to the server via a dedicated channel, where they are decoded and input into the ASR backend. Studies have shown that DSR generally performs better than NSR [1] because, in the latter model, speech is processed for optimal perceptual quality and this does not necessarily result in optimal recognition performance [9].

Various schemes for compressing the ASR features have been proposed in the literature. Digalakis et al. in [10] evaluated the use of uniform and non-uniform scalar quantizers as well as product code vector quantizers for compressing Mel frequency-warped cepstral coefficients (MFCCs) between 1.2 and 10.4 kbps. They concluded that split vector quantizers achieved word error rates (WER) similar to that of scalar quantizers while requiring less bits. Also, scalar quantizers with non-uniform bit allocation performed better than those with uniform bit allocation. Ramaswamy and Gopalakrishnan [11] investigated the application of tree-searched multistage vector quantizers with one-step linear prediction operating at 4 kbps. Transform coding, based on the discrete cosine transform (DCT), was investigated in [12] at 4.2 kbps and in [13] which used a two-dimensional DCT. The ETSI DSR standard [14] uses split vector quantizers to compress the MFCC vectors at 4.4 kbps. Srinivasamurthy et al. in [9] exploited correlation across consecutive MFCC features by using a DPCM scheme followed by entropy coding.

Even though vector quantizers generally give better recognition performance at a lesser bit-rate, they are not scalable in bit-rate when compared with scalar quantizer-based coding schemes, such as DPCM and transform coders. In other words, the vector quantizer is designed to operate at a specific bit-rate only and will need to be re-trained for other bit-rates. *Bit-rate scalability* is a desirable feature in DSR applications, since one may need to adjust the bit-rate adaptively, depending on the network conditions. For instance, if the communications network is heavily congested, then it may be more acceptable to sacrifice some recognition performance by operating at a lower bit-rate in order to offset long response times. In addition to this, the computational complexity of vector quantizers can be quite high, when compared with scalar quantizer-based schemes. Therefore, in this paper, we propose a fixed-rate block quantization scheme for DSR applications[1] that is computationally simpler than vector quantizers, is scalable in bit-rate, and leads to a

[1] While we are quantising MFCC features for DSR, this quantizer can also be applied in a CELP speech coder to be used in NSR.

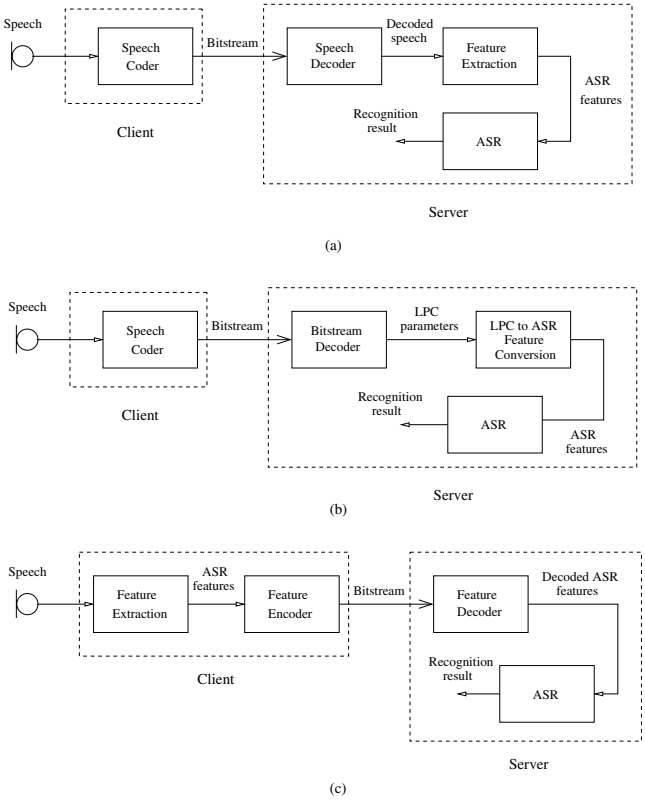

Figure 1: Client-server-based speech recognition modes: (a) Speech-based network speech recognition (NSR); (b) Bitstream-based network speech recognition; (c) Distributed speech recognition (DSR)

more graceful degradation in recognition performance when compared with other scalar quantizer-based schemes.

2. Description of the proposed coder

This coding scheme is based on the one proposed by Subramaniam and Rao in [15] for the coding of speech line spectral frequencies (LSF), where a Gaussian mixture model (GMM) is used to parametrically model the probability density function (PDF) of the source and block quantizers are then designed for each Gaussian mixture component. In [16], we proposed a modified scheme which used vectors formed from p concatenated frames, in order to exploit interframe correlation. Therefore, if the length of the MFCC frame is n, then the dimensions of the vectors processed will be np. MFCCs are calculated frame-wise from speech and there is considerable overlap between successive frames. Generally, there will be high correlation between consecutive frames [9]. Therefore, we have chosen to use multi-frame GMM-based block quantization ($p = 5$) to exploit this correlation. For more details on this coding scheme as well as its memory and computational requirements, the reader is referred to [15].

2.1. Quantizer training

The PDF model and Karhunen-Loéve transform (KLT) orthogonal matrices are the only static and bit-rate-independent parameters of the GMM-based block quantizer. These only need to be calculated once (training) and stored at the client encoder and server decoder. The bit allocations for different bit-rates (described in Section 2.2.1) can be calculated 'on-the-fly' based on the PDF model by both client and server. Hence this scheme is bit-rate scalable [15].

The PDF model, which is in the form of a GMM, is initialized by applying the K-means algorithm on the training vectors where m clusters[2] are produced, each represented by a mean, $\boldsymbol{\mu}$, a covariance matrix, $\boldsymbol{\Sigma}$, and cluster weight, c. These form the initial parameters for the GMM estimation procedure. Using the Expectation-Maximization (EM) algorithm, the maximum-likelihood estimate of the parametric model is computed iteratively until the log likelihood converges, where a final set of means, covariance matrices, and weights are produced.

An eigenvalue decomposition (EVD) is calculated for each of the m covariance matrices, producing m eigenvalues, $\{\boldsymbol{\lambda}_i\}_{i=1}^{m}$, and eigenvectors. The eigenvectors form the rows of the orthogonal transformation matrix, $\boldsymbol{K}$, of the KLT.

2.2. Encoding process

2.2.1. Bit allocation

Assuming that there are b_{tot} bits available for coding each vector (for an average bit-rate of b_{tot}/np bps), these need to allocated to each of the block quantizers for each cluster. The number of bits, b_i, allocated to the block quantizer of cluster i, is given by [15]:

$$2^{b_i} = 2^{b_{tot}} \frac{(c_i \Lambda_i)^{\frac{np}{np+2}}}{\sum_{i=1}^{m} (c_i \Lambda_i)^{\frac{np}{np+2}}}, \quad \text{for } i = 1, 2, \ldots, m \tag{1}$$

where [15]:

$$\Lambda_i = \left(\prod_{j=1}^{np} \lambda_{i,j} \right)^{\frac{1}{np}} \quad \text{for } i = 1, 2, \ldots, m \tag{2}$$

Then for each block quantizer, the traditional high resolution formula from [18][3] is used to distribute the b_i bits to each of the vector components:

$$b_{i,j} = \frac{b_i}{np} + \frac{1}{2} \log_2 \frac{\lambda_{i,j}}{\left(\prod_{j=1}^{np} \lambda_{i,j} \right)^{\frac{1}{np}}} \quad \text{for } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, np \tag{3}$$

2.2.2. Minimum distortion block quantization

Fig. 2 shows a diagram of minimum distortion block quantization. It consists of m independent Gaussian block quantizers, BQ_i, each with their own orthogonal matrix, $\boldsymbol{K}_i$, and bit allocations, $\{b_{i,j}\}_{j=1}^{np}$. The following coding process is described in [15].

To quantize a vector, $\boldsymbol{x}$, using a particular cluster i, the cluster mean, $\boldsymbol{\mu}_i$, is first subtracted and its components decorrelated using the orthogonal matrix, $\boldsymbol{K}_i$, for that cluster. The variance of each component is then normalized by the standard deviation to produce a decorrelated, mean-subtracted, and normalized-variance vector, $\boldsymbol{z}_i$:

$$\boldsymbol{z}_i = \frac{\boldsymbol{K}_i(\boldsymbol{x} - \boldsymbol{\mu}_i)}{\boldsymbol{\sigma}_i} \tag{4}$$

[2] The terms 'cluster' and 'mixture component' are used interchangeably in this paper

[3] While the greedy bit allocation algorithm of Riskin [19] leads to a more optimal solution, the high resolution formula in [18] is computationally more efficient.

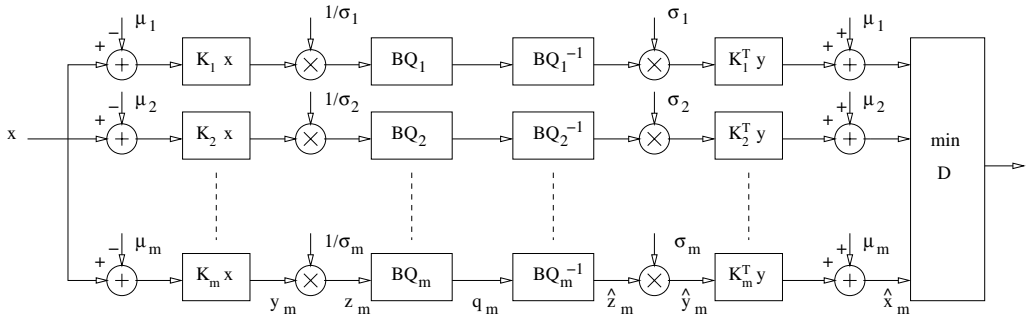

Figure 2: Block diagram of the GMM-based block quantizer (BQ - block quantizer)

These are then quantized using a set of np Gaussian Lloyd-Max scalar quantizers as described in [18] with their respective bit allocations producing indices, $\boldsymbol{q}_i$. These are decoded to give the approximated normalized vector, $\widehat{\boldsymbol{z}}_{\boldsymbol{i}}$, which is multiplied by the standard deviation and correlated again by multiplying with the transpose, $\boldsymbol{K}_i^T$, of the orthogonal matrix. The cluster mean is then added back to give the reconstructed vector, $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$.

$$\widehat{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{K}^T \boldsymbol{\sigma}_i \widehat{\boldsymbol{z}}_{\boldsymbol{i}} + \boldsymbol{\mu}_i \tag{5}$$

The distortion between this reconstructed vector and original is then calculated, $d(\boldsymbol{x} - \widehat{\boldsymbol{x}}_i)$. The above procedure is performed for all clusters in the system and the cluster, k, which gives the *minimum distortion* is chosen:

$$k = \underset{i}{\operatorname{argmin}}\, d(\boldsymbol{x} - \widehat{\boldsymbol{x}}_i) \tag{6}$$

In the case of coding MFCC vectors, we use the mean-squared-error (MSE) as the distortion measure for selecting the appropriate block quantizer.

3. Experimental setup

We have evaluated the recognition performance of scalar quantization as well as single frame and multi-frame GMM-based block quantization using the publicly available HTK 3.2 software on the Aurora-2 database [17]. We have limited the focus of this work on recognition performance versus bit-rate in clean and matched conditions only. Training was done on clean data and testing was performed on the clean data of test set A. The four word recognition accuracies are then averaged to give the final score for the specific coding scheme. The parameters for the HMM models are the same as those stated in [17].

The ETSI DSR standard Aurora frontend [14] was used for the MFCC feature extraction. As a slight departure from the ETSI DSR standard, we have used 12 MFCCs (excluding the zeroth cepstral coefficient, c_0, and logarithmic frame energy, $\log E$) as the feature vectors to be quantized. This was done to maintain consistency in the coding scheme as c_0 and $\log E$ are sensitive to changes in recording level of a speech utterance and are generally coded independently [14, 12, 11]. Sinusoidal liftering was applied to the MFCCs using the following window function, $w(n)$:

$$w(n) = 1 + \frac{L}{2} \sin\left(\frac{\pi n}{L}\right) \quad \text{where } n = 1, 2, \ldots, L \tag{7}$$

where L is the feature length. After decoding the 12 MFCC features and concatenating them with their corresponding delta and acceleration coefficients, the final feature vector dimension for the ASR system is 36.

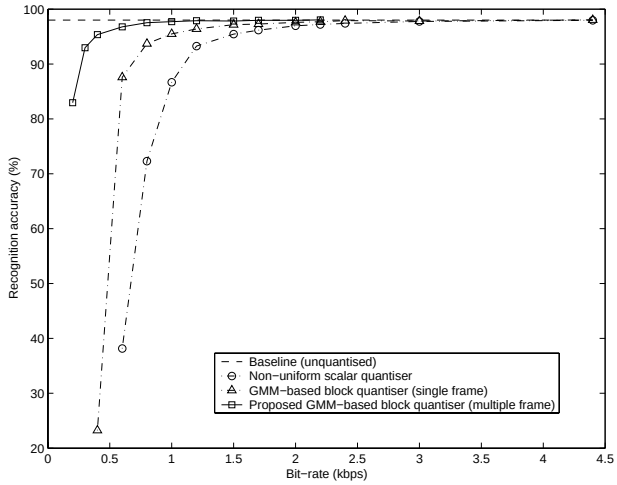

Figure 3: Average word recognition accuracy as a function of bit-rate

For the scalar quantization experiment, each MFCC was quantized using a Gaussian Lloyd-Max scalar quantizer whose bit allocation was calculated using the high resolution formula of (3). We have chosen this method over the WER-based greedy algorithm of [10] because of its computational simplicity.

In the training of the GMM-based block quantizer, 20 iterations of the EM algorithm were used to generate a 16 cluster GMM. For the multi-frame GMM-based block quantizer, 5 MFCC feature vectors were concatenated to form vectors of dimension 60.

4. Results and discussion

Table 1 shows the average recognition accuracy over a range of bit-rates for the different coding schemes. These results are also plotted in Fig. 3. The baseline average recognition accuracy, where uncoded MFCC features were used, is 98.01%. We can observe that the scalar quantiser (SQ) incurs a small degradation (about 1%) in recognition performance, over the baseline, at bit-rates of 2 kbps and above. However, the recognition performance drops dramatically below 1.2 kbps. The single frame GMM-based block quantiser (GMM-1) achieves higher recognition scores than the scalar quantiser for all bit-rates and gradually degrades until about 800

Table 1: Average word recognition accuracy as a function of bit-rate for different coding schemes (baseline accuracy = 98.01%)

	Recognition Accuracy (in %)		
Bit-rate (kbps)	SQ	GMM-1	GMM-5
0.2			82.94
0.3			92.96
0.4		23.25	95.36
0.6	38.17	87.59	96.78
0.8	72.31	93.70	97.52
1.0	86.68	95.49	97.71
1.2	93.27	96.40	97.89
1.5	95.45	97.17	97.83
1.7	96.17	97.28	97.96
2.0	96.97	97.58	97.97
2.2	97.21	97.70	98.04
2.4	97.40	97.90	
3.0	97.76	97.83	
4.4	97.96	98.04	

bps, where it starts to drop sharply. Its advantages over the scalar quantiser lie in more accurate modelling of the source as well as the exploitation of correlation within each frame by the KLT. The proposed multi-frame GMM-based block quantiser (GMM-5) achieves the highest recognition accuracy of all three coding schemes, with negligible degradation ($< 1\%$) in recognition accuracy over the baseline, at 800 bps. Even below 800 bps, the performance of the proposed coder degrades gracefully with a WER of 7% at 300 bps. This demonstrates that adjacent frames exhibit a high degree of correlation and the joint coding of these frames leads to improved coding efficiency. It can be observed from the results that this improved efficiency is accompanied by high recognition performance.

5. Conclusion

In this paper, we have proposed the use of the multi-frame GMM-based block quantizer for the coding of MFCC features in DSR applications. The strengths of this coding scheme are its computational simplicity when compared with vector quantizers, bit-rate scalability, and graceful degradation of recognition performance at very low bit-rates via effective exploitation of intraframe and interframe correlation. Because the PDF model and transformation matrices are independent of bit-rate, these static parameters can be stored on the encoder and decoder and the bit-rate can be adjusted 'on-the-fly', depending on network conditions. This coding scheme exhibits negligible degradation of 1% (WER of 2.5%) in recognition performance over the baseline system at 800 bps and 5% (WER of 7%) at 300 bps.

6. References

[1] I. Kiss, "A comparison of distributed and network speech recognition for mobile communication systems", in *Proc. ICSLP*, 2000.

[2] H.G. Hirsch, "The influence of speech coding on recognition performance in telecommunication networks", in *Proc. ICSLP*, Denver, USA, Sept. 1998.

[3] J.M. Huerta and R.M. Stern, "Speech recognition from GSM codec parameters", in *Proc. ICSLP*, Vol. 4, 1998, pp.1463–1466.

[4] H.K. Kim and R.V. Cox, "A bitstream-based front-end for wireless speech recognition on IS-136 communications system", *IEEE Trans. Speech Audio Processing*, Vol. 9, No. 5, pp. 558–568, July 2001.

[5] B.T. Lilly and K.K. Paliwal, "Effect of speech coders on speech recognition performance", in *Proc. ICSLP*, Vol. 4, 1996, pp. 2344–2347.

[6] B. Raj, J. Migdal and R. Singh, "Distributed speech recognition with codec parameters", in *Proc. ASRU*, Trento, Italy, Dec. 2001.

[7] J. Turunen and D. Vlaj, "A study of speech coding parameters in speech recognition", in *Proc. Eurospeech*, 2001, pp. 2363–2366.

[8] A. Gallardo-Antolin, F. Diaz-de-Maria and F. Valverde-Albacete, "Recognition from GSM digital speech", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1998, pp. 1443–1446.

[9] N. Srinivasamurthy, A. Ortega and S. Narayanan, "Efficient scalable encoding for distributed speech recognition", submitted to *IEEE Trans. Speech and Audio Processing*, 2003. Available: http://biron.usc.edu/~snaveen/papers/Scalable_DSR.pdf

[10] V.V. Digalakis, L.G. Neumeyer and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the world wide web", *IEEE J. Select. Areas Commun.*, Vol. 17, No. 1, pp. 82–90, Jan 1999.

[11] G.N. Ramaswamy and P.S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1998, pp. 977–980.

[12] I. Kiss and P. Kapanen, "Robust feature vector compression algorithm for distributed speech recognition", in *Proc. Eurospeech*, 1999, pp. 2183–2186.

[13] Q. Zhu and A. Alwan, "An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Vol. 1, Aug 2001, pp. 113-116.

[14] "Speech processing, transmission and quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", Tech. Rep. Standard ES 201 108, European Telecommunications Standards Institute (ETSI), April 11 2000.

[15] A.D. Subramaniam and B.D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies", *IEEE Trans. Speech Audio Processing*, Vol. 11, No. 2, pp. 130–142, Mar. 2003.

[16] K.K. Paliwal and S. So, "Multiple frame block quantisation of line spectral frequencies using Gaussian mixture models", to appear in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2004.

[17] H.G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", *ISCA ITRW ASR2000*, Paris, France, Sept. 2000.

[18] J.J.Y. Huang and P.M. Schultheiss, "Block quantization of correlated Gaussian random variables", *IEEE Trans. Commun. Syst.*, Vol. CS-11, pp. 289–296, Sept. 1963.

[19] E.A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm", *IEEE Trans. Inform. Theory*, 37(2), pp. 400–402, 1991.