Quantization of Speech Features: Source Coding

Stephen So¹ and Kuldip K. Paliwal²

¹ Griffith University, Griffith School of Engineering, Signal Processing Laboratory, s.so@griffith.edu.au

Abstract. In this chapter, we describe various schemes for quantizing speech features to be used in distributed speech recognition (DSR) systems. We have analyzed the statistical properties of MFCCs that are most relevant to quantization, namely the correlation and probability density function shape, in order to determine the type of quantization scheme that would be most suitable for quantizing them efficiently. We also determine empirically the relationship between mean squared error and recognition accuracy in order to verify that quantization schemes, which minimize mean squared error, are also guaranteed to improve the recognition performance. Furthermore, we highlight the importance of noise robustness in DSR and describe the use of a perceptually weighted distance measure to enhance spectral peaks in vector quantization. Finally, we present some experimental results on the quantization schemes in a DSR framework and compare their relative recognition performances.

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 incorporating automatic speech recognition (ASR) technology into mobile communication systems. Speech recognition can facilitate consumers in performing common tasks, which have traditionally been accomplished via buttons and/or pointing devices.

Distributed speech recognition (DSR) is a mode of client-server-based ASR, where speech features are extracted on the client device and then transmitted to the server, which performs the recognition task, as shown in Fig. 1. In order to motivate the need for quantization, let us calculate the bitrate that is required to transmit *uncoded* feature vectors. If feature vectors of 13 Mel frequency-warped cepstral coefficients (MFCCs) are extracted at a frame rate of 100 Hz and that each MFCC is represented as a 32 bit floating point value, then the required bitrate is 41.6 kbps. As we shall see later on, current state-of-the-art quantization schemes used in DSR can operate at bitrates as low as 300 bps.

² Griffith University, Griffith School of Engineering, Signal Processing Laboratory, k.paliwal@griffith.edu.au



Fig. 1. Block diagram of a typical distributed speech recognition system

In this chapter, we are interested in the lossy coding of feature vectors for DSR applications. The ultimate aim is to quantize feature vectors using the least amount of bits, while maintaining a recognition performance that is as close as possible to that of ASR. Note that when we use the term *ASR performance*, we are referring to the recognition performance achieved when no lossy coding has been applied to the feature vectors, as opposed to *DSR performance*, where feature vectors have been coded in a lossy fashion. It is reasonable to assume that, using the same features, DSR performance will generally be less than and upper bounded by the ASR performance, hence the latter serves as a useful baseline for evaluating quantization schemes.

This chapter is divided into five sections. In the first section, we will review some basic concepts of source coding and quantization as well as outline some quantization schemes that will be evaluated later in the chapter. In the second section, we examine the statistical properties of the MFCC feature vectors as well as determine the relationship between mean squared error and recognition accuracy. In the third section, we present a brief review of the literature on the topic of quantizing feature vectors. Following this, we will present some results of recent quantization schemes that we have investigated in our laboratory (So and Paliwal 2005; So and Paliwal 2006). We then conclude the chapter in the final section.

2 Quantization Schemes

2.1 Brief Introduction to Quantization Theory

Source coding schemes can be broadly classified into two categories: lossless and lossy coding. While lossless coding incurs no loss of information (that is, the decoded output data is exactly the same as the input data), the amount of compression is limited by the Shannon entropy of the data (Gersho and Gray 1992). Examples of lossless coding schemes (often referred to as *entropy coders*) include Huffman coding, arithmetic coding, runlength encoding, etc.



Feature Encoder

Fig. 2. Block diagram of the 'Feature encoder' in Fig. 1, showing the quantization scheme and binary encoder

It is common for an entropy coder to be cascaded on the output of a lossy coder to further reduce the bitrate (Gray and Neuhoff 1998). An example of this is in the JPEG image coder, where the output coefficients of the lossy scalar quantization stage are coded using a runlength encoder and a Huffman coder (Wallace 1991). While it is possible to apply entropy coding on the output of the quantization schemes discussed in this chapter to reduce the bitrate further, various complications arise, such as the resulting bitrate being variable over time. Therefore, buffering is often required to handle the variable bitrates, which adds to the complexity of the overall DSR system.

On the other hand, lossy coding schemes have no constraints on the amount of compression that can be achieved, hence they are often more useful in scenarios where channel capacity is low and limited. The bitrate of lossy coding schemes can be made fixed, thus removing the requirement for buffering. The challenge with lossy coding schemes is minimizing the distortion given a fixed bitrate, or given an allowed and fixed distortion, minimizing the bitrate required – this is often referred to as the *rate-distortion tradeoff*.

Quantization is a fundamental process for information reduction in lossy coding schemes and is generally the source of information loss. It is defined as the mapping of individual (scalar) or a vector of input samples to a codebook of a finite number of *codewords*. Each codeword has a unique binary word or index associated with it so each input sample is substituted with this binary word before transmission. The mapping is done in such a way that the distortion incurred by substituting the input sample by its corresponding codeword is minimized. The input samples may be quantized individually (referred to as *scalar quantization*), or as vectors (referred to as *vector quantization*). Figure 2 shows where the quantization scheme 'fits' in the DSR feature encoder.

The rate-distortion (RD) efficiency of any quantizer is influenced by the properties of the signal source, such as statistical dependencies (otherwise known as memory) and the probability density function (PDF) (Makhoul, Roucos *et al.* 1985). Furthermore, it has been shown that vector quantizers always have a better RD efficiency than scalar quantizers, and therefore are optimal quantizers (Lookabaugh and Gray 1989). The properties of the speech features used in DSR will be discussed in

the following subsections. However, before we move on, we will present popular distortion measures that have been used in speech processing as well as describe the quantization schemes that will be evaluated later in the chapter.

2.2 Distortion Measures for Quantization in Speech Processing

It is important to define the distortion measure to be used in quantizers as different applications may require the minimization of an error calculation that incorporates some signal-based or perceptual properties in order to improve the overall fidelity. The simplest distortion measure that is commonly used in the coding literature is *mean squared error* (MSE), d_{MSE} , which is defined below:

$$d_{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = E[(\mathbf{x} - \hat{\mathbf{x}})^{T}(\mathbf{x} - \hat{\mathbf{x}})]$$
(1)

In this equation, $E[\bullet]$ is the expectation operator, x and \hat{x} are the input vector and quantized vector, respectively, and \bullet^T is the transpose operator. The error contribution of each vector component is weighted the same.

Weighted distortion measures are often used to perform *quantization noise shaping*, which can improve the overall fidelity by exploiting signal-based properties. For example, in speech coding applications, line spectral frequency (LSF) vectors can be quantized using a weighted mean square error, where the error contributions of each LSF are non-uniformly weighted based on the relative spectral power at that particular frequency (Paliwal and Atal 1993). For components that have a higher weighting, the quantization error will be less. This weighted mean squared error (WMSE) can be expressed as:

$$d_{WMSE}(\mathbf{x}, \hat{\mathbf{x}}) = E[(\mathbf{x} - \hat{\mathbf{x}})^T W (\mathbf{x} - \hat{\mathbf{x}})]$$
(2)

In this equation, W is a square diagonal weighting matrix whose diagonal elements consist of the relative weightings of each vector component.

Another common distortion measure that is used for evaluation in speech coding is the *logarithmic spectral distortion* (this is often simply referred to as *spectral distortion*). It is defined as the root mean squared error between the log power spectral density estimates of the original and quantized frame of speech:

$$d_{SD} = \sqrt{\frac{1}{F_s} \int_0^{F_s} [10 \log_{10} P(f) - 10 \log_{10} \hat{P}(f)]^2 df}$$
(3)



Fig 3. Block diagram of scalar quantization of vectors, with mean removal, variance normalization and bit allocation.

In Eq. 3, F_s is the sampling frequency, P(f) and $\hat{P}(f)$ are the power spectral density estimates of the input and quantized speech frame, respectively. It can be shown that the MSE distortion measure in the cepstral domain is equivalent to the spectral distortion (Rabiner and Juang 1993).

Other distortion measures that have been used in speech processing include the Itakura-Saito distortion, Itakura distortion, COSH distance, etc. (Rabiner and Juang 1993). For distributed speech recognition, the quantization distortion measure should be somewhat correlated to the desired performance metric – recognition accuracy. We will discuss this further in Section 3.4. Because of their relatively low computational complexity, we will mostly focus on MSE-based distortion measures as these need to be computed multiple times in quantization schemes such as VQ.

2.3 Scalar Quantization

The simplest quantizer is the *scalar quantizer* (SQ), where input samples are mapped individually to scalar codewords, which are also referred to as code-points or reproduction values (Gersho and Gray 1992). The number of reproduction values or quantization levels, *n*, is given by $n = 2^b$, where *b* is the number of bits.

For input samples that have a non-uniform probability density function, such as Gaussian or Laplacian, it has been shown that non-uniform scalar quantizers incur less distortion than uniform scalar quantizers, where quantization levels are uniform-ly spaced (Max 1960). The quantization levels for Gaussian and other arbitrary distributions (with zero-mean and unity variance) have been reported in the coding literature (Max 1960; Paez and Glisson 1972; Lloyd 1982). The input samples

should have zero-mean and normalized variance before quantization, as shown in Fig. 3.

When quantizing a vector of input samples using scalar quantizers, we need to allocate the bit budget among the individual SQs. For example, if the vector dimensionality is n and the bitrate is fixed at b bits/sample, then a total of nb bits need to be allocated to the n SQs. The objective is to determine the best bit allocation such that the quantization distortion is minimized. We discuss two methods for bit allocation in scalar quantization: high resolution-based optimization (HRO) and the greedy-based heuristic algorithm.

In HRO bit allocation, which was first presented in relation to block quantization (Huang and Schultheiss 1963), the average distortion incurred by the overall scalar quantization scheme is expressed in terms of the high resolution approximation of the non-uniform scalar quantizer:

$$d_{avg} = \frac{1}{n} \sum_{i=1}^{n} K \sigma_i^2 2^{-2b_i}$$
(4)

In this equation, *n* is the vector dimensionality, *K* is a constant which varies for different PDFs (for Gaussian PDFs, $K = \frac{\pi\sqrt{3}}{2}$), σ_i^2 is the variance of the *i*th vector component, and b_i is the number of bits allocated to the *i*th scalar quantizer. This expression is to be minimized using the fixed bitrate constraint,

$$b_{tot} = \sum_{i=1}^{n} b_i \tag{5}$$

We are then left with the following bit allocation formula (for the full derivation, see (Huang and Schultheiss 1963)):

$$b_{i} = \frac{b_{tot}}{n} + \frac{1}{2} \log_{2} \frac{\sigma_{i}^{2}}{\left[\prod_{i=1}^{n} \sigma_{i}^{2}\right]^{\frac{1}{n}}}$$
(6)

Let us consider an example of scalar quantizing vectors of dimension 4 using a total of 20 bits, given the following variances: $\sigma_i^2 = \{2, 30, 10, 52\}$. Using Eq. 6, we calculate a bit allocation of $b_i = \{3.634, 5.587, 4.7948, 5.984\}$ bits. We note that, firstly, more bits have been allocated to vector components with higher variances; and secondly, the formula gives fractional (and even negative in some cases) bit allocations. One may truncate these fractional bit allocations though this generally leads to a total bitrate that is less than the target. A method is presented in (Paliwal and So 2005) for handling fractional bit allocations so that more of the bit budget is

utilized. A further constraint that enforces the b_i to be always positive may also be applied to the optimization process (Segall 1976).

The greedy-based heuristic algorithm for allocating bits is simpler than the HRO algorithm and is more readily applicable to vectors with non-standard PDFs, where deriving closed-form expressions may be difficult or impossible. Allocation is performed one bit at a time for each vector component, with the one resulting in the largest drop in quantization distortion to be selected to receive the bit. The process continues until all bits have been allocated, where the resulting solution may only be locally optimal. Greedy-based heuristic bit allocation has been investigated in DSR in the literature (Digalakis, Neumeyer *et al.* 1999).

2.4 Block Quantization

In *block quantization*, also known as *transform coding*, an orthogonal linear transformation P, whose columns consist of the basis vectors, is applied to a zero-mean input vector, \mathbf{x} , before scalar quantization (Huang and Schultheiss 1963):

$$\mathbf{y} = \mathbf{P}^T \mathbf{x} \tag{7}$$

where *y* is the transformed vector containing the transform coefficients, $\{y_i\}_{i=1}^n$. The inverse linear transformation is expressed as:

$$x = Py \tag{8}$$

The covariance matrix of the transformed vectors is given by:

$$\Sigma_{y} = E[yy^{T}]$$
$$= E[P^{T}x(P^{T}x)^{T}]$$
$$= P^{T}E[xx^{T}]P$$
$$= P^{T}\Sigma_{x}P$$

When scalar quantizing input samples, the statistical dependencies between these samples are not exploited and this leads to wasted bits and thus inefficient quantization. In block quantization, the linear transformation serves to decorrelate the samples before scalar quantization, which will improve the coding efficiency. The correlation is 'added' back in the decoding stage via the inverse transformation of Eq. 8.

The decorrelating transformation also tends to pack the energy or variance into the first few coefficients. When using the HRO bit allocation formula of Eq. 6, the skewed variance distribution of the transformed coefficients will cause more bits to be allocated to the scalar quantizers of the first few coefficients. Typical transformations used in coding include the Karhunen-Loève transform (KLT) and the discrete cosine transform (DCT).

2.4 Vector Quantization

The basic definition of a vector quantizer Q of dimension n and size K is a mapping of a vector from n dimensional Euclidean space, \mathfrak{R}^n , to a finite set, C, containing K reproduction *codevectors*:

$$Q:\mathfrak{R}^n\to C \tag{9}$$

where $C = \{y_i; i \in I\}$ and $y_i \in \Re^n$. Associated with each reproduction codevector is a partition of \Re^n , called a region or cell, $S = \{S_i; i \in I\}$.

The most popular form of vector quantizer is the *Voronoi* or *nearest neighbour* vector quantizer (Gersho and Gray 1992), where for each input source vector x, a search is done throughout the entire codebook to find the nearest codevector y_i , which has the minimum distance:

$$\mathbf{y}_{i} = Q[\mathbf{x}] \quad \text{if } d(\mathbf{x}, \mathbf{y}_{i}) < d(\mathbf{x}, \mathbf{y}_{i}) \quad \text{for all } i \neq j \tag{10}$$

where d(x, y) is the distortion measure between the vectors, x and y. Generally, the most common distortion measure used in vector quantizers is the MSE.

The VQ codebook is designed using a large number of training vectors, which are representative of the set of vectors that will be quantized by the VQ. The iterative Linde-Buzo-Gray (LBG) algorithm (Linde, Buzo *et al.* 1980) is applied to the training vectors and the resulting *K* centroids or codevectors constitute the VQ codebook. The bitrate of the vector quantizer is $\log_2 K$ bits/vector.

Though the *unconstrained VQ* (that is, the VQ codebook has no structural constraints) is theoretically the optimal quantizer that one can design, its computational complexity and memory requirements may become prohibitive at high bitrates. Furthermore, designing a high bitrate VQ codebook requires a large amount of training data. Therefore, the application of unconstrained VQ is often constrained to low bitrates, while structurally-constrained forms, such as multistage, split, and treestructured VQ are used when higher bitrates are required. Constrained VQs sacrifice rate-distortion performance for lower computational and memory requirements.

2.5 GMM-Based Block Quantization

The GMM-based block quantizer (Subramaniam and Rao 2003) is an improved version of the Gaussian block quantizer (Huang and Schultheiss 1963). Rather than assume the PDF of the input vectors to be Gaussian, Gaussian mixture models (GMMs) are used to approximate the PDF and each mixture component is quantized using a Gaussian block quantizer. These modifications result in better RD performance as the GMM-based block quantizer is designed to match the PDF more closely, assuming that there is minimal overlap between the mixture components.



Fig 4. PDF estimation and bit allocation from training data

Compared with vector quantizers, the GMM-based block quantizer has the advantages of: fixed computational and memory requirements that are independent of the bitrate; and *bitrate scalability*, where any bitrate can be used without the need to redesign the codebook (Subramaniam and Rao 2003). Bitrate scalability is a desirable feature in DSR applications, since one may need to adjust the bitrate adaptively, depending on the network conditions (So and Paliwal 2006).

This quantization scheme can be broken down into three stages: PDF estimation, bit allocation and minimum distortion block quantization. Each stage will be described in the following subsections.

PDF Estimation using Gaussian Mixture Models

The PDF model and Karhunen-Loève transform (KLT) orthogonal matrices are the only static and bitrate-independent parameters of the GMM-based block quantizer. These only need to be calculated once during the training stage and stored at the client encoder and server decoder. The bit allocations for different bitrates can be calculated 'on-the-fly' using the common PDF model stored on both client and server. The PDF estimation procedure is shown in Fig. 4.

The PDF model, G, as a mixture of multivariate Gaussians, $N(x; \mu, \Sigma)$, can be expressed as:

$$G(\mathbf{x} \mid \mathbf{M}) = \sum_{i=1}^{m} c_i N(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$
(11)

$$\boldsymbol{M} = [\boldsymbol{m}, \boldsymbol{c}_1, \dots, \boldsymbol{c}_m, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_m]$$
(12)

$$N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$
(13)

where x is a source vector, m is the number of mixture components, and n is the dimensionality of the vector space. c_i , $\mu_i \Sigma_i$ are the weight, mean, and covariance matrix of the *i*th mixture component, respectively.

The parametric model, M, is initialized by applying the LBG algorithm (Linde, Buzo *et al.* 1980) on the training vectors where m mixture components are produced, each represented by a mean or centroid, μ , a covariance matrix, Σ , and a mixture component weight, c. These form the initial parameters for the GMM estimation procedure. Using the expectation-maximization (EM) algorithm (Dempster, Laird *et al.* 1977), the maximum-likelihood estimate of the parametric model is computed iteratively and a final set of means, covariance matrices, and weights are produced.

An eigenvalue decomposition (EVD) is calculated for each of the m covariance matrices. The eigenvectors form the rows of the orthogonal transformation matrix, K, of the KLT.

Bit Allocation

Assuming there are a total of b_{tot} bits available for quantizing each vector, these need to be allocated to each of the block quantizers of each mixture component in an optimal fashion. Using Lagrangian minimization (Subramaniam and Rao 2003), the following formula is derived:

$$2^{b_{i}} = 2^{b_{int}} \frac{(c_{i}\Lambda_{i})^{\frac{n}{n+2}}}{\sum_{i=1}^{m} (c_{i}\Lambda_{i})^{\frac{n}{n+2}}} \text{ for } i = 1, 2, ..., m$$
(14)
$$\Lambda_{i} = \left[\prod_{j=1}^{n} \lambda_{i,j}\right]^{\frac{1}{n}}$$
(15)

In Eqs. 14 and 15, $\lambda_{i,j}$ is the *j*th eigenvalue of mixture component *i* and b_i is the number of bits allocated to the block quantizer of mixture component *i*.

Once bits have been allocated to the block quantizer of each mixture component, these need to be further allocated to the scalar quantizers within the block quantizer. The bit allocation was presented in Section 2.3 and the formula for allocating bits is given by Eq. 6.

Quantization of Speech Features: Source Coding



Fig. 5. Minimum distortion block quantization (BQ - block quantizer)

Minimum Distortion Block Quantization

Figure 5 shows the minimum distortion block quantization stage, whose operation is described in more detail in (Subramaniam and Rao 2003). At first glance, it can be seen to consist of *m* independent block quantizers, BQ_i , each with their own orthogonal matrix, K_{i} , and bit allocations, $\{b_{i,j}\}_{j=1}^{n}$. A vector, x, is quantized m times and the kth block quantizer is chosen such that it incurs the least distortion.

$$k = \arg\min_{i} d(\mathbf{x}, \hat{\mathbf{x}}_{i})$$
(16)

3 Quantization of ASR Feature Vectors

3.1 Introduction and Literature Review

So far, we have only discussed quantization and the various schemes in general with no reference made to quantizing ASR feature vectors. In this section, we discuss the task of quantizing ASR feature vectors as well as examine some statistical properties that may affect the quantization and recognition performance. We will also examine the performance of the DSR system in the presence of background noise. Unless otherwise specified, we will be mostly focusing on Mel frequency-warped cepstral coefficients (MFCCs) (Davis and Mermelstein 1980) as the ASR feature set.

Various schemes for quantizing the ASR features have been proposed in the literature. Digalakis et al. (1999) evaluated the use of uniform and non-uniform scalar quantizers as well as product code vector quantizers for coding MFCCs at rates of between 1.2 and 10.4 kbps. They used the greedy-based bit allocation algorithm for the scalar quantizers, where the component, which resulted in the largest improvement in recognition performance, was chosen to receive the allocated bit. They concluded that split vector quantizers achieved word error rates (WER) similar to that of scalar quantizers while requiring fewer bits. A bitrate of 2 kbps was the required bitrate for split vector quantization to achieve ASR recognition performance.

Also scalar quantizers with non-uniform bit allocation performed better than those with uniform bit allocation.

In (Ramaswamy and Gopalakrishnan 1998), the authors investigated the application of tree-searched multistage vector quantizers (MSVQ) with first-order linear prediction operating at a bitrate of 4 kbps. The current MFCC feature vector was subtracted from the previous quantized frame to give a residual vector. The first 12 coefficients of the residual vector were then quantized using a two-stage MSVQ, while the last coefficient, c_0 , was scalar quantized. Their system achieved near identical recognition performance as the ASR recognition performance, with only minor degradation.

Transform coding, based on the DCT, was investigated in (Kiss and Kapanen 1999) at a bitrate of 4.2 kbps. In this scheme, feature vectors of dimension 14 (13 MFCCs plus the energy coefficients, c_0 and log E) were processed. For each cepstral coefficient, eight temporally consecutive coefficients were grouped together and processed by the DCT, which exploited temporal correlation. The energy coefficient was encoded separately.

In (Zhu and Alwan 2001), 12 successive MFCC frames were stacked together to form a block of 12×12 and a two-dimensional DCT was applied. Zonal sampling was performed, where a fraction of the lowest energy components was set to zero and the remaining coefficients were scalar quantized and entropy coded. The advantage of this scheme compared to that of (Kiss and Kapanen 1999) is that both within-frame and across-frame correlation is exploited by the 2D-DCT. Noise-robust feature sets, such as peak isolated MFCCs (MFCCP) (Strope and Alwan 1997) and variable frame-rate peak isolated MFCCs (VFR_MFCCP) (Zhu and Alwan 2000) were also tested. Their results showed that, firstly, the DSR recognition performance always performed slightly worse than the ASR recognition performance at all signal-to-noise (SNR) levels. Secondly, the quantized noise-robust features at 624 bps resulted in recognition accuracies that even surpassed the ASR performance at low SNRs.

The ETSI DSR standard (2003) uses split vector quantizers to compress the MFCC vectors at 4.4 kbps. Feature vectors of dimension 14 (13 MFCCs and log E) are split into pairs of subvectors, with the energy parameters, c_0 and log E belonging to the same pair. A weighted MSE distortion measure is used for the energy parameter subvector.

In (Srinivasamurthy, Ortega *et al.* 2006), correlation across consecutive MFCC features was exploited by a differential pulse coded modulation (DPCM) scheme followed by entropy coding. Their scheme is a scalable one, where the bitstream is embedded. That is, a coarsely quantized *base layer* is transmitted. If higher recognition performance is required, the client can transmit further *enhancement layers*, which are combined with the base layer by the server to obtain higher quality features.

13

3.2 Statistical Properties of MFCCs

The statistical properties of the MFCC vectors have a direct influence on the ratedistortion performance of any quantization scheme. According to Makhoul (1985), these properties are:

- 1. linear dependency (i.e. correlation);
- 2. non-linear dependency;
- 3. probability density function shape; and
- 4. dimensionality (i.e. quantizing vectors is more efficient than scalars).

We will investigate properties 1 and 3 of MFCC vectors in the following sub-section. In particular, the correlation across successive vectors will be examined as this property is exploited by interframe schemes such as multiframe/matrix and prediction-based quantizers.

Correlation within MFCC Vectors (Intraframe Dependencies)

We examine the amount of correlation between cepstral coefficients within a feature vector by computing the covariance matrix of MFCCs from the training speech set of the Aurora-2 database (Hirsch and Pearce 2000). The MFCCs consist of 13 cepstral coefficients, $\{c_i\}_{i=0}^{12}$. The log energy coefficient *log E*, which is often concatenated with the MFCC feature set in ASR, has not been included. Rather than presenting a 13 × 13 matrix of coefficients, we have plotted the absolute value of the covariance coefficients in Fig. 6. Because of the large difference in magnitude of the variance of c_0 compared with those of the other cepstral coefficients, we have applied a square root operation to the covariance coefficients to compress the dynamic range. Therefore, the coefficients on the diagonal represent the standard deviation of each cepstral coefficient rather than the variance.

We can see that a large percentage of the energy is contained in the zeroth cepstral coefficient, c_0 . Recall that the final stage of MFCC computation comprises a discrete cosine transform (DCT), which tends to compact most of the energy into the zeroth cepstral coefficient or DC component. In addition, most of the off-diagonal covariance coefficients have low magnitude, which indicates that the cepstral coefficients are weakly correlated with each other – apart from c_0 , where the cross-variance with the other cepstral coefficients appears to be higher. This suggests that the other cepstral coefficients $\{c_i\}_{i=1}^{12}$ contain some information of the zeroth cepstral coefficient. Hence, in most speech recognition systems, c_0 is not included in the feature set.

Because the efficiency of scalar quantization is generally optimal when the vector components are not correlated (which is the basis of block quantization), the covariance statistics of MFCCs (shown in Fig. 6) suggest that directly scalar quantizing the MFCCs may not be optimal. In which case, a further transform (such as the KLT) may be required to remove the remaining correlation and henceforth improve the rate-distortion performance.



Fig. 6. Graphical representation showing the absolute value of the covariance coefficients of MFCCs within a single vector with compressed dynamic range (log energy is not included)

This improvement will be become apparent when comparing the results between the scalar quantizer and the block quantizer.

Correlation across Successive MFCC Vectors (Interframe Dependencies)

In order to examine the correlation across successive MFCC vectors, we concatenate these vectors to form higher dimensional vectors and compute the covariance matrix of this new vector set. Any linear dependencies between MFCCs in successive vectors will be shown by large off-diagonal coefficients in the corresponding rows and columns of the covariance matrix. Figure 7 is similar to Fig. 6, where the covariance matrix is graphically represented in a three dimensional representation. We also present the graphical covariance matrix representation for two, three, four, and five concatenated MFCC vectors in order to show the amount of correlation between MFCCs across these successive vectors. As before, the log energy coefficient has not been included and an absolute value followed by a square root operation has been applied to all covariance coefficients in order to compress the dynamic range.

Looking at Fig. 7(a), where two vectors have been concatenated together, we notice a large number of off-diagonal covariance coefficients that have a large magnitude, which indicates a high degree of correlation between the MFCCs across successive frames.



Fig. 7. Graphical representation showing the coefficients of the covariance matrix of MFCCs within a multiple successive vectors with compressed dynamic range: (a) two vectors; (b) three vectors; (c) four vectors; (d) five vectors

This is to be expected, as the speech frames used to compute the MFCCs are highly overlapped. When we look at the covariance coefficients for three, four, and five vectors, in Figs. 7(b), (c), and (d), we notice greater numbers of off-diagonal elements with large magnitude. Therefore, it is expected that quantization schemes, which exploit memory across multiple successive, will be more efficient in the rate-distortion sense, than memoryless schemes.

We should point out that this method of vector concatenation does not capture all of the dependencies. For example, if we represent four successive MFCC vectors as x_1, x_2, x_3, x_4 , then concatenating them will produce: $[x_1, x_2], [x_3, x_4]$. The covariance matrix will capture the dependencies between MFCCs in both x_1 and x_2 and between MFCCs in both x_3 and x_4 , but not the dependences between x_2 and x_3 .



Fig. 8. Graphical representation showing the prediction coefficients from a single-step linear prediction of MFCC vectors (c_0 and log E are represented as cepstral coefficient 13 and 14, respectively)

As a further method of capturing the correlation that exists across successive frames, we apply a *single-step* backward prediction analysis using the covariance method over the MFCC feature vector set to compute prediction coefficients. Both the energy coefficients, c_0 and log E have been included. Up to 10 past vectors were used in the analysis. The closer the prediction coefficients are to unity, the higher the degree of correlation between any MFCC vector and a past vector. Fig. 8 shows a graphical representation of the prediction coefficients for each cepstral coefficient. We can see that consecutive vectors (past vector number equal to one) are highly correlated as is shown by the prediction coefficients being closest to unity. The coefficients decrease in value as vectors further away in the past are used to predict the current vector, with some MFCCs decreasing faster than others. It is interesting to point out that the energy coefficients across 10 frames are highly correlated. This observation suggests that the energy coefficients could be efficiently quantized using prediction-based schemes.

Probability Density Functions of MFCCs

The probability density function (PDF) of MFCCs are particularly important when we consider scalar quantization-based schemes. Fig. 9 shows the probability density function (PDF) estimates of the MFCCs in addition to the log E coefficient.



Fig. 9. Probability density function estimates of MFCCs

The PDFs of the MFCCs, apart from c_0 and log E, resemble unimodal Gaussians, which suggests that they are amenable to non-uniform scalar quantization optimized for Gaussian sources as well as block quantization. This is to be expected as the MFCCs were formed from linear combinations of vector components during the DCT operation. According to the central limit theorem, as the dimension of the vectors increases, the distributions of the transform coefficients approach a Gaussian (Chen and Smith 1977). In contrast, the c_0 and log E coefficients possess a bimodal distribution, which suggests that custom-designed scalar quantizers would be needed here.

We conclude this section on the statistical properties of MFCCs by noting the differences in the statistics of the energy coefficients (c_0 and log E) when compared with those of $c_1...c_{12}$, in terms of the correlation and PDF. It is for this reason that the energy coefficients are often quantized independently from the rest of the cepstral coefficients. Because of this, the issue of bit allocation arises. That is, how much of the bit budget should be allocated for quantizing energy coefficients in order to maximize the recognition performance? The majority of the quantization schemes reported in the literature have arbitrarily allocated bits to the energy coefficients, rather than utilising a formula obtained from constrained minimization. The problem is that it is not entirely clear how much impact quantization errors in the energy coefficients. In order to isolate the uncertainty associated with energy coefficient quantization as well as to present a simple and consistent bit allocation framework, we have performed all DSR experiments where the energy coefficients are not included as part of the MFCC feature set.

3.3 Use of Cepstral Liftering for MFCC Variance Normalization

The variances of each MFCC are shown in Fig. 10. The variances of c_0 and log E (not shown in Fig. 10), are 2530 and 260, respectively. The non-uniform variance distribution of the MFCCs is a result of the energy-packing characteristics of the discrete cosine transform. It is also well known that the lower order cepstral coefficients are particularly sensitive to undesirable variations caused by factors such as transmission, speaker characteristics, vocal efforts, etc. (Juang, Rabiner *et al.* 1987).

According to the HRO bit allocation formula for scalar quantization in Eq. (6), bits are allocated to vector components on the basis of variance, in order to minimize the mean squared error. This can be seen in the first row of Table 1, which shows the number of bits that are allocated to each MFCC, using HRO bit allocation. Because c_1 has the highest variance, it has been allocated the most number of bits.



Fig. 10. Variances of MFCCs (c_0 and log E are not included)



Fig. 11. Variances of MFCCs after cepstral liftering (c_0 and log E are not included)

	Total bits	c_1	c_2	<i>c</i> ₃	c_4	c_4	c ₆	<i>C</i> ₇	C ₈	C9	c_{10}	c_{11}	<i>c</i> ₁₂
Without liftering	15	3.1	2.4	1.9	1.6	1.3	1.1	0.9	0.7	0.7	0.6	0.4	0.3
With liftering	15	2.4	2.3	2.2	2.2	2.0	1.8	1.5	1.3	1.0	0.5	-0.4	-2

 Table 1. Number of bits allocated to each MFCC with and without the application of cepstral liftering (computed using Eq. (6))



Fig. 12. Lifter window function of Eq. (16)

From a quantization point of view, where the mean squared error between the original and reconstructed MFCC vectors is minimized, finely quantizing the first few MFCCs makes sense since they have higher variance. As will be shown in the next section, the relationship between MSE and recognition accuracy is monotonic and non-linear. However, if the operating bitrate is low, there may be a shortage of bits to allocate to the important middle-order MFCCs.

If the shortage of bits that is due to a low operating bitrate, is found to cause a performance degradation, then one may normalize the variances of the MFCCs so that the bit allocation is not too highly skewed. This normalization can be done via the use of *liftering*, which performs 'filtering' in the cepstral domain. Cepstral liftering was a technique that was investigated in the literature to improve the recognition performance (Paliwal 1982), where cepstral coefficients were linearly weighted. Another method of cepstral liftering (Juang, Rabiner *et al.* 1987) uses the following sinusoidal lifter window function:

$$w(n) = 1 + \frac{L}{2} \sin\left[\frac{\pi n}{L}\right]$$
(16)



Fig. 13. Relationship between average recognition rate and average MSE

where L is the dimensionality of the MFCCs. This window function is plotted in Fig. 12, where we can see an emphasis on the middle order cepstral coefficients. The effect of the liftering operation on the MFCC variances and the bit allocation are shown in Fig. 11 and Table 1, respectively, where bits are allocated more uniformly to the middle order MFCCs. In our experiments, we have used cepstral liftering for the purpose of variance normalization. Further work is needed to determine the benefits that it may provide to the recognition performance as well as noise-robustness in a DSR scenario. This is in light of the results presented in (Paliwal 1999), where cepstral liftering on MFCCs was shown to improve the noise robustness for dynamic time warping-based speech recognizers, which use Euclidean distance measures.

3.4 Relationship Between the Distortion Measure and Recognition Performance

All quantization schemes attempt to minimize the error between the original and quantized samples. For instance, the HRO bit allocation formula of Eq. (6) for scalar quantizing vector components was obtained from a constrained minimization of the average MSE. In vector quantization, the codebook vector that minimizes the distortion is selected.

The direct application of these quantization schemes to distributed speech recognition readily assumes that decreasing the MSE between the original and quantized MFCC features will guarantee that the degradation in recognition performance due to the quantization decreases as well. We will validate this assumption by applying unconstrained vector quantization on MFCCs at varying bitrates, measuring the average MSE and recognition rates for each bitrate. Figure 13 shows the average recognition rate plotted against the average MSE incurred by the vector quantizer.

22 Stephen So and Kuldip K. Paliwal



Fig. 14. Extraction of logarithmic filterbank energies from speech

We can see from Fig. 13 that the recognition rate appears to decrease monotonically as the average MSE increases. Therefore, this shows that a quantization scheme that minimizes the MSE is also guaranteed to improve the recognition accuracy. Furthermore, we note that it is a non-linear relationship, where if the average MSE was large, a decrease in quantization distortion leads to a larger improvement in recognition rate than if the MSE were low.

3.5 Improving Noise Robustness: Perceptual Weighting of Filterbank Energies

Noise-robustness is an important consideration in DSR since the user at the client end will mostly be immersed in various environmental sounds. It is well known in the ASR literature that noise has a detrimental effect on the recognition performance when using conventional MFCC features. Much of the current work in ASR research involves finding speech features that are robust to the effects of noise. These speech features can be used in DSR as well.

Using a two-dimensional discrete cosine transform coder, Zhu and Alwan (2001) improved the robustness of DSR to noise by using peak-isolated MFCCs (MFCCPs). MFCCPs are derived by applying half-wave rectification to the spectrum recon-

structed from a bandpass liftered cepstral vector (Strope and Alwan 1997). They are robust to noise because of the preservation and emphasis of power spectral peaks, whose frequency locations are known to be important for the discrimination of vowels. The idea is that accuracy in the location of spectral peaks is more important than the location of spectral valleys.

Another method of exploiting this idea is to quantize the logarithmic filterbank energies (LFBEs) rather than the MFCCs themselves (So and Paliwal 2005). The advantage of working with LFBEs is their correspondence with the power spectrum. That is, a strong peak in the power spectrum would generally lead to a large LFBE coefficient in the same critical band. On the other hand, the frequency location information of this spectral peak is not readily available in the MFCC representation as each MFCC consists of a linear combination of all LFBEs. By quantizing the LFBEs, we can apply noise-shaping techniques to quantize LFBEs that correspond to spectral peaks more finely than those that correspond to spectral valleys. The disadvantage of using LFBE vectors is that they have a higher dimensionality than MFCC vectors.

In order to achieve quantization noise shaping, we apply a perceptually-weighted distance measure to vary the emphasis of the quantization, which can easily be incorporated into a vector quantizer (So and Paliwal 2005). The weighted distance measure $d_w(E, \hat{E})$ between the original LFBE vector E and the LFBE \hat{E} is defined as:

$$d_{w}(\boldsymbol{E}, \hat{\boldsymbol{E}}) = \sum_{i=1}^{n} \left[w_{i}(E_{i} - \hat{E}_{i}) \right]^{2}$$
(17)

where *n* is the vector dimensionality, w_i is the weight of the *i*th component, E_i and \hat{E}_i are the *i*th component of the original and code-vector, respectively. In order to emphasize a vector component, E_i , such that it is quantized more finely, the weight w_i should be made larger. In the LFBE vector quantizer, it is desirable to emphasize the LFBEs that represent the spectral peaks. Therefore, w_i is set to be a scaled version of the FBE, e^{E_i} :

$$w_i = \left[e^{E_i}\right]^r \tag{18}$$

Through experimentation, we have found 0.5 to be a good value for r.

4 Experimental Results

4.1 ETSI Aurora-2 Distributed Speech Recognition Task

The purpose of the ETSI Aurora-2 experiment is to provide a common framework for evaluating noise-robust speech recognition systems. It consists of a clean speech

database, a noise database, a standard MFCC-based frontend, and scripts for performing the various training and test sets. The recognition engine that is used is the HMM Toolkit (HTK) software (Young, Evermann *et al.* 2002).

The TIDigits database (Leonard 1984) forms the basis of the clean speech database, where the original 20 kHz speech was downsampled to 8 kHz and filtered using the frequency characteristic of ITU G.712 (300--3400 Hz). Aurora-2 also provides a database of eight background noises, which were deemed to be commonly encountered in real-life operating conditions for DSR. These noises were recorded at the following places (Hirsch and Pearce 2000):

- Suburban train (subway)
- Crowd of people (babble)
- Car
- Exhibition hall (exhibition)
- Restaurant
- Street
- Airport
- Train station

This noise is added to the filtered clean speech at various SNRs to simulate noise corruption.

There are two training modes: training with clean speech only and training with clean and noisy (multicondition) speech (Hirsch and Pearce 2000). In multicondition training, the noises added are subway, babble, car, and exhibition. When training with clean speech only, the best recognition performance is achieved in matched conditions, i.e. when testing with clean speech as well. However, when the speech to be tested has background noise, then multicondition training is desirable, as it includes the distorted speech in the training data (Hirsch and Pearce 2000).

For the testing, there are three test sets, known as test set A, B, and C. In test set A and B, 4004 test utterances from the TIDigits database are divided into four subsets of 1001 utterances each and four different types of noises are added to each subset at varying levels of SNRs (∞ , 20, 15, 10, 5, 0, -5 dB). Therefore, there are a total of 4 × 7 = 28 recognition accuracies reported in test set A and B. In test set C, only two subsets of 1001 utterances and two noises are used, giving a total of 14 recognition accuracies.

In test set A, the subway, babble, car, and exhibition noises are added to each subset and these are the same noises used in multicondition training, hence test set A evaluates the system in matched conditions. In test set B, the other four noises, namely restaurant, street, airport, and train station, are used instead. Because these noises were not present in the multicondition training, then test set B evaluates the system in mismatched conditions (mismatched noise). Test set C contains two utterance subsets only (of the four) with the noises, subway and street, added. Both the speech and noise are filtered using the MIRS frequency characteristic before they are added, hence test set C evaluates the system in mismatched conditions (mismatched conditions (mismatched conditions (mismatched conditions frequency characteristic before they are added, hence test set C evaluates the system in mismatched conditions (mismatched frequency characteristic) (Hirsch and Pearce 2000).

25

Whole word HMMs are used for modelling the digits with the following parameters:

- 16 states per word (with 2 dummy states at beginning and end);
- left-to-right topology without skips over states;
- 3 Gaussian mixtures per state; and
- diagonal covariance matrices.

4.2 Experimental Setup

We have evaluated the recognition performance of various quantization schemes version 3.2.1 of the HMM Toolkit (HTK) software. Training was done on clean data only (no multicondition training) and testing was performed using test set A. In order to see the recognition performance as a function of bitrate, we focus on the results of testing on *clean speech*, where the four word recognition accuracies for each type of noise are averaged to give the final score for the specific quantization scheme. In addition to this, the effect of different types of noise at varying levels of SNR on the recognition performance is also investigated at the bitrates of 1.2 kbps and 0.6 kbps for each quantization scheme.

The ETSI DSR standard Aurora frontend (2003) was used for the MFCC feature extraction. MFCCs are extracted at a frame rate of 100 Hz. 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. We have applied the cepstral liftering technique (Juang, Rabiner *et al.* 1987) to the MFCC vectors. Cepstral mean subtraction (CMS) is applied to the decoded 12 MFCC features, which are concatenated with their corresponding delta and acceleration coefficients, giving the final feature vector dimension of 36 for the ASR system. The HTK parameter type is MFCC_D_A_Z. The baseline average recognition accuracy or ASR accuracy using unquantized MFCC features derived from clean speech is 98.0 %.

4.3 Non-Uniform Scalar Quantization Using HRO Bit Allocation

For the scalar quantization experiment, each MFCC was quantized using a nonuniform Gaussian Lloyd-Max scalar quantizer whose bit allocation was calculated using the HRO bit allocation formula of Eq. (6). We have chosen this method over the WER-based greedy algorithm (Digalakis, Neumeyer *et al.* 1999) because of its computational simplicity and this allows us to scale any bitrate with ease. Table 2 shows the average recognition accuracy of the non-uniform scalar quantizer. It can be seen that the accuracy decreases linearly in the range of 4.4 to 1.2 kbps and drops rapidly below this range.

Bitrate (kbps)	Average recognition accuracy (in %)
0.6	38.2
0.8	72.3
1.0	86.7
1.2	93.3
1.5	95.5
1.7	96.2
2.0	97.0
2.2	97.2
2.4	97.4
3.0	97.8

26 Stephen So and Kuldip K. Paliwal

4.4

 Table 2. Average DSR word recognition accuracy as a function of bitrate for non-uniform scalar quantizer (ASR accuracy = 98.0%)

98.0

Bitrate (kbps)	Average recognition accuracy (in %)
0.4	76.9
0.6	91.8
0.8	95.7
1.0	96.9
1.2	97.0

Table 3. Average DSR word recognition accuracy as a function of bitrate for the unconstrained vector quantizer (ASR accuracy = 98.0%)

4.4 Unconstrained Vector Quantization

An unconstrained, full-search vector quantizer was used to quantize single MFCC frames. The distance measure used was MSE. In terms of minimizing quantization distortion, the vector quantizer is considered the optimum coding scheme, hence it will serve as an informal upper recognition bound for single frame quantization. Table 3 shows the average recognition accuracies at several bitrates.

When comparing with Table 2, we can see that the superior rate-distortion efficiency of the vector quantizer translates to better recognition rates as well. For example, at 600 bps, which corresponds to 6 bits in total for quantizing 12 coefficient MFCC vectors, the recognition rate for the vector quantizer is 53.6% higher than that for the scalar quantizer. With such a small bit budget, the scalar quantizer cannot allocate bits to some MFCCs, thus in the decoding, they would simply be replaced by the mean value. On the other hand, the vector quantizer codebook, which contains 64 code-vectors, exploits linear and non-linear dependencies between the MFCCs, matches the joint PDF, and uses optimal quantization cell shapes (Lookabaugh and Gray 1989).

Bitrate (kbps)	Average recognition accuracy (in %)
0.3	8.1
0.4	23.3
0.6	87.6
0.8	93.7
1.0	95.5
1.2	96.4
1.5	97.2
1.7	97.3
2.0	97.6
2.2	97.7
2.4	97.9
3.0	97.8
4.4	98.0

27

 Table 4. Average DSR word recognition accuracy as a function of bitrate for the GMM-based block quantizer with 16 mixture components (ASR accuracy = 98.0%)

4.5 GMM-Based Block Quantization

Table 4 shows the average recognition accuracies for the GMM-based block quantizer with 16 mixture components. We can see that for this quantization scheme, the recognition accuracy decreases gracefully to about 800 bps. Comparing it with Table 2, we notice higher recognition accuracies in the GMM-based block quantizer, which may be attributed to better PDF matching as well as the use of a decorrelating transformation. At 600 bps, the GMM-based block quantizer is 49.4% better than the scalar quantizer. However, it is not as high as the recognition performance achieved with the vector quantizer at 600 bps (Table 3). This is consistent in the rate-distortion sense since the vector quantizer should be the optimum single-frame quantizer. However, in practice, the vector quantizer suffers from high computational complexity, while the GMM-based block quantizer has fixed requirements as well as possessing the feature of bitrate scalability.

4.6 Multi-frame GMM-Based Block Quantization

The multi-frame GMM-based block quantizer is similar to the matrix quantizer (Tsao and Gray 1985). Five successive MFCC frames are concatenated to form a vector of dimension 60 and these larger vectors are then quantized. Table 4 shows the average word recognition accuracy of the 16 mixture component, five frame multi-frame GMM-based block quantizer for different bitrates.

Bitrate (kbps)	Average recognition accuracy (in %)
0.2	82.9
0.3	93.0
0.4	95.4
0.6	96.8
0.8	97.5
1.0	97.7
1.2	97.9
1.5	97.8
1.7	98.0
2.0	98.0

Table 5. Average word recognition accuracy as a function of bitrate for the multi-frame GMM-based block quantizer with 16 mixtures and 5 frames (ASR accuracy = 98.0%)

It can be observed that this quantizer achieves an accuracy that is close to the unquantized, baseline system at 1 kbps or 10 bits/frame, which is half the bitrate of the single-frame GMM-based block quantizer. For bitrates lower than 600 bps, the performance gradually rolls off.

In terms of quantizer distortion, the multi-frame GMM-based block quantizer generally performs better as more frames are concatenated together because interframe memory can be exploited by the KLT. Furthermore, because the dimensionality of the vectors is high, the block quantizer operates at a higher rate.

Compared with the results of the single frame GMM-based block quantizer in Table 4, the multi-frame scheme does not suffer from a dramatic drop in recognition accuracy at low bitrates. Unlike the single frame scheme, where there was a shortage of bits to distribute among mixture components, the multi-frame GMM-based block quantizer is able to provide enough bits, thanks to the increased dimensionality of the vectors. For example, at 300 bps, a 16-mixture component, single frame GMM-based block quantizer has a total bit budget of 3 bits. On the other hand, a 16-mixture component, five-frame scheme has a total bit budget of 15 bits. Therefore, the multi-frame GMM-based block quantizer can operate at lower bitrates while maintaining good recognition performance.

The multi-frame GMM-based block quantizer also outperforms the vector quantizer since the latter is only a single frame scheme. As we have seen previously, successive MFCC frames are highly correlated with each other so it is expected that quantization schemes that exploit multiple frame dependencies will perform much better in the rate-distortion sense. The disadvantage of this scheme is the inherent delay that is introduced.



Figure 15. Word recognition accuracy for speech corrupted with noise at varying SNRs (in dB) at 1.2 kbps using the perceptually weighted vector quantizer on LFBEs (PWVQ-LFBE) (solid line represents the ASR accuracy; squares represent PWVQ-LFBE and crosses represent VQ-MFCC): (a) corrupted with subway noise; (b) corrupted with babble noise; (c) corrupted with car noise; (d) corrupted with exhibition noise

4.7 Perceptually-Weighted Vector Quantization of Logarithmic Filterbank Energies

We can see from Fig. 15 that the proposed perceptually weighted vector quantization scheme operating on logarithmic filterbank energies (PWVQ-LFBE) is more robust to noise than the unweighted vector quantization of MFCCs (VQ-MFCC). At SNRs of 10 and 15 dB, the PWVQ-LFBE scheme achieves up to 6 to 10% improvement over VQ-MFCC. This may be attributed to the use of the weighted distance measure to emphasize the spectral peaks. However, for low SNRs, the PWVQ-LFBE scheme fails to improve the noise robustness, when compared with VQ-MFCC. Furthermore, this quantization scheme appears to be bounded by the ASR recognition accuracy (shown as the solid line in Fig. 15). We should point out that higher bitrates were not investigated due to computational constraints.

5 Conclusion

In this chapter, we have described a series of quantization schemes for coding MFCC feature vectors that are to be used for distributed speech recognition. These include the scalar quantizer, vector quantizer, perceptually weighted vector quantizer and GMM-based block quantizer. These quantization schemes have been described in detail in the coding literature but their application to quantizing MFCC feature vectors has been a relatively recent development. It is important to note that the objective measure in DSR that is to be optimized is the recognition accuracy, rather than the mean squared error. Therefore, quantization in the context of DSR deserves further investigation.

We have discussed the statistical properties of MFCCs that are relevant to quantization. In particular, we have shown that successive MFCC vectors are highly correlated with each other. Because of this property, multi-frame and predictive quantization schemes should perform more efficiently. In relation to the energy coefficients (c_0 and log E), which were shown to possess different statistical properties, we concluded that they should be quantized independently from the rest of the cepstral coefficients. We have also shown via empirical results that the recognition rate increases monotonically as MSE decreases. That is, optimizing quantizers to minimize the MSE, in general, should guarantee an improvement in recognition rate. However, the relationship is a non-linear one.

Next, we presented a brief review of the distributed speech recognition literature, where various schemes for quantizing MFCCs were investigated. The Aurora-2 database used for evaluating the performance of our MFCC quantization schemes as well as the parameters for the recognition task were described in detail. Following this, we presented our results on MFCC quantization in a DSR framework using non-uniform scalar quantization with HRO bit allocation, vector quantization, and single-frame as well as multi-frame GMM-based block quantization. For clean speech, the multi-frame GMM-based block quantizer achieved the best recognition at lower bitrates, exhibiting a negligible 1% degradation (word error rate of 2.5%) in recognition performance over the ASR accuracy at 800 bps and 5% degradation (word error rate of 7%) at 300 bps. Unlike vector quantization schemes, the multi-frame GMM-based block quantizer is scalable in bitrate and has a complexity that is independent of bitrate.

We also looked at the performance of vector quantization of MFCCs derived from noise corrupted speech at various SNR levels and compared this with the perceptually-weighted vector quantizer (PWVQ). Rather than quantizing MFCCs, the PWVQ works with logarithmic filterbank energies (LFBEs). The non-linearly weighted distance measure allows for the shaping of quantization noise, putting more emphasis on spectral peaks so that they are quantized more finely. We showed that this scheme improves noise-robustness for medium SNRs (10 to 15 dB) over the vector quantization of MFCCs.

References

- (2003). Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms. Tech. Rep. Standard ES 201 108 v1.1.3, European Telecommunications Standards Institute (ETSI).
- Chen, W. and Smith, C. H. (1977). "Adaptive Coding of Monochrome and Color Images." IEEE Trans. Commun. **COM-25**(11): 1285-1292.
- Davis, S. B. and Mermelstein, P. (1980). "Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences." IEEE Trans. Acoust., Speech, Signal Processing ASSP-28(4): 357-366.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." J. Roy. Stat. Soc. 39: 1-38.
- Digalakis, V. V., Neumeyer, L. G. and Perakakis, M. (1999). "Quantization of Cepstral Parameters for Speech Recognition over the World Wide Web." IEEE J. Select. Areas Commun. 17(1): 82-90.
- Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Massachusetts.
- Gray, R. M. and Neuhoff, D. L. (1998). "Quantization." IEEE Trans. Inform. Theory 44(6): 2325-2383.
- Hirsch, H. G. and Pearce, D. (2000). The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems Under Noisy Conditions. ISCA ITRW ASR2000, Paris, France.
- Huang, J. J. Y. and Schultheiss, P. M. (1963). "Block Quantization of Correlated Gaussian Random Variables." IEEE Trans. Commun. CS-11: 289-296.
- Juang, B. H., Rabiner, L. R. and Wilpon, J. G. (1987). "On the Use of Bandpass Liftering for Speech Recognition." IEEE Trans. Acoust., Speech, Signal Processing 1: 597-600.
- Kiss, I. and Kapanen, P. (1999). Robust Feature Vector Compression Algorithm for Distributed Speech Recognition. European Conference on Speech Communication and Technology.
- Leonard, R. G. (1984). A Database for Speaker-Independent Digit Recognition. Proc. IEEE. Int. Conf. Acoust., Speech, Signal Processing.
- Linde, Y., Buzo, A. and Gray, R. M. (1980). "An Algorithm for Vector Quantizer Design." IEEE Trans. Commun. **28**(1): 84-95.
- Lloyd, S. P. (1982). "Least Square Quantization in PCM." IEEE Trans. Inform. Theory IT-28(2): 129-137.
- Lookabaugh, T. D. and Gray, R. M. (1989). "High-Resolution Quantization Theory and the Vector Quantizer Advantage." IEEE Trans. Inform. Theory **35**(5): 1020-1033.
- Makhoul, J., Roucos, S. and Gish, H. (1985). "Vector Quantization in Speech Coding." Proc. IEEE **73**: 1551-1588.
- Max, J. (1960). "Quantizing for Minimum Distortion." IRE Trans. Inform. Theory IT-6: 7-12.
- Paez, M. D. and Glisson, T. H. (1972). "Minimum Mean-Squared-Error Quantization in Speech PCM and DPCM System." IEEE Trans. Commun. COM-20: 225-230.
- Paliwal, K. K. (1982). "On the Performance of the Quefrency-Weighted Cepstral Coefficients in Vowel Recognition." Speech Communication 1: 151-154.
- Paliwal, K. K. (1999). Decorrelated and Liftered Filterbank Energies for Robust Speech Recognition. European Conf. Speech Communication Technology, Budapest, Hungary.

- Paliwal, K. K. and Atal, B. S. (1993). "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame." IEEE Trans. Speech Audio Processing 1(1): 3-14.
- Paliwal, K. K. and So, S. (2005). "A Fractional Bit Encoding Technique for the GMM-Based Block Quantization of Images." Digital Signal Processing **15**(3): 435-446.
- Rabiner, L. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey.
- Ramaswamy, G. N. and Gopalakrishnan, P. S. (1998). Compression of Acoustic Features for Speech Recognition in Network Environments. IEEE Int. Conf. Acoust., Speech, Signal Processing.
- Segall, A. (1976). "Bit Allocation and Encoding of Vector Sources." IEEE Trans. Inform. Theory **IT-22**(2): 162-169.
- So, S. and Paliwal, K. K. (2005). Improved Noise-Robustness in Distributed Speech Recognition via Perceptually-Weighted Vector Quantisation of Filterbank Energies. European Conference on Speech Communication and Technology, Lisbon, Portugal.
- So, S. and Paliwal, K. K. (2006). "Scalable Distributed Speech Recognition Using Gaussian Mixture Model-Based Block Quantisation." Speech Communication 48: 746-758.
- Srinivasamurthy, N., Ortega, A. and Narayanan, S. (2006). "Efficient Scalable Encoding for Distributed Speech Recognition." Speech Communication **48**(8): 888-902.
- Strope, B. and Alwan, A. (1997). "A Model of Dynamic Auditory Perception and its Application to Robust Word Recognition." IEEE Trans. Speech Audio Processing 5(2): 451-464.
- Subramaniam, A. D. and Rao, B. D. (2003). "PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequencies." IEEE Trans. Speech Audio Processing **11**(2): 130-142.
- Tsao, C. and Gray, R. M. (1985). "Matrix Quantizer Design for LPC Speech using the Generalized Lloyd Algorithm." IEEE Trans. Acoust., Speech, Signal Processing 33: 537-545.
- Wallace, G. K. (1991). "The JPEG Still Picture Compression Standard." Communications of the ACM 34(4): 30-44.
- Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P. (2002). *The HTK Book (for HTK Version 3.2.1)*. Cambridge University Engineering Department.
- Zhu, Q. and Alwan, A. (2000). On the Use of Variable Frame Rate Analysis in Speech Recognition. IEEE Int. Conf. Acoust., Speech, Signal Processing.
- Zhu, Q. and Alwan, A. (2001). An Efficient and Scalable 2D DCT-Based Feature Coding Scheme for Remote Speech Recognition. IEEE Int. Conf. Acoust., Speech, Signal Processing.