Towards Efficient and Scalable Speech Compression Schemes for Robust Speech Recognition Applications*

N. Srinivasamurthy, A. Ortega
Integrated Media Systems Center, Dept of EE-Systems, USC
[snaveen,ortega]@sipi.usc.edu

Q. Zhu, A. Alwan
Electrical Engineering Department, UCLA
[qifeng,alwan]@icsl.ucla.edu

Abstract

This paper presents a scheme for distributed automatic speech recognition. A Hidden Markov Model (HMM)-based speech recognition system with a Mel Frequency Cepstral Coefficients (MFCC) front end was used in the evaluation. The goal was to achieve good recognition performance while compressing the MFCC feature vectors. Compression rates and recognition performance for both a digit and an alphabet database are reported. Compared to a scheme of recognizing speech encoded by low bit rate encoders, and previously-reported schemes, our method can achieve good recognition performance with bitrates lower than 1 kbps, using low encoding complexity. The encoding algorithms developed are scalable, allowing bitrate and recognition performance trade-offs, and can be combined with unequal error protection or prioritization to allow graceful degradation of performance in the presence of channel errors.

1 Introduction

In this paper we study speech compression for distributed speech recognition, where speech is first acquired and then transmitted to a remote recognition engine. An example of this situation would be that of a user employing a portable wireless device to access a remote speech-driven application. In this situation recognition may be too expensive computationally to be performed at the portable device. Other examples include distributed web applications where a single centralized recognition database is kept for security or scalability reasons [1].

Given the reduced channel bit rate available in typical applications (especially mobile ones) compression will be required to transmit speech to the remote recognizer. However, current speech coding techniques focus on preserving the perceptual quality of the speech. Thus when recognition, rather than playback, is the ultimate objective, it is desirable to modify the coding techniques so that they maximize recognition performance rather than preserve perceptual quality.

In our approach we assume that Hidden Markov Model (HMM) based recognizers are being used with a Mel Frequency Cepstral Coefficient (MFCC) [2] front end. As in previous work [3, 4] we assume the client extracts the MFCCs and then compresses the coefficients to transmit them to the recognition engine. We present a complete compression algorithm based on scalar quantization, linear prediction [4], entropy coding and coefficient pruning. Unlike previous work [3, 4], our coding algorithm is *scalable*, that is, it is possible to reduce the coding rate (for example if the channel conditions worsen and additional channel coding is needed), at the cost of some decrease in the recognition performance. Scalability is achieved by changing the quantization step size to reduce the number of coefficients transmitted. In addition to scalability, the proposed technique achieves similar recognition performance at lower rates, and with significantly lower complexity, than previously proposed approaches.

2 Compression of MFCCs

The encoding algorithms presented in this section were developed assuming that the feature vectors used by the speech recognizer are 12 MFCCs derived from every frame of the speech utterance. However, these results can be easily extended to cases where the number of MFCCs in a frame is not 12, or when derivatives are used along with the MFCCs.

MFCCs computed from speech are represented by floating point numbers (32 bits precision is typical). Clearly, it is to be expected that reductions in the precision through quantization will be possible without affecting the recognition performance. In addition, the MFCCs are derived from speech utterances that

*This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.

0-7803-6536-4/00/$10.00 (c) 2000 IEEE

have been segmented using overlapping Hamming windows. Due to this overlap it is reasonable to expect that MFCC sets corresponding to adjacent frames will exhibit high correlation. We exploit this correlation by using linear prediction, where a given MFCC in a frame is predicted from the corresponding MFCC in one or more past frames. Single-step prediction seems a reasonable choice given that the time overlap occurs only between adjacent frames, and indeed our experiments showed that the gain in applying multistage prediction was limited. Thus, in what follows we consider only single step linear prediction, where each MFCC is predicted only from the corresponding MFCC in the previous frame. Note that while quantization of MFCCs for speech recognition has been previously considered in [3] and [4], our approach provides better performance at similar or lower complexity. For example, [3] uses scalar quantization (uniform and non-uniform) and product vector quantization (VQ) techniques but does not use either entropy coding or linear prediction. In [4] a one step linear prediction is used along with a 2-Stage VQ that achieves a fixed rate of 4 kbps. This approach lacks scalability and has a significantly more complex encoder than the approaches we present here. Finally, neither [3] nor [4] employed entropy coding, which, as will be shown, can improve further the compression gains at low bitrates.

To quantize the MFCCs (or the prediction errors after linear prediction) we use two different scalar quantization techniques, namely, entropy constrained scalar quantization (ECSQ) [5] and uniform scalar quantization (USQ). In the ECSQ approach the quantizer is designed by minimizing, for each input in the training set, a cost function of the form $\mathbf{C} = \mathbf{D} + \lambda * \mathbf{R}$ where $\mathbf{D}$ and $\mathbf{R}$ are the distortion and rate, respectively, and λ is a Lagrange multiplier, which is a non-negative real number that is used to control the rate-distortion trade-off. Given that the statistics are different we designed different quantizers for each MFCC. Moreover, different quantizers were used depending on whether the coefficient was coded directly, or the error after linear prediction was coded instead. As a simpler alternative, we used USQ to quantize the prediction errors after linear prediction. The same quantization step size can be used for all the coefficients if the prediction errors are divided first by their corresponding standard deviation, which can be computed during training.

One of our main goals in designing this system was to introduce scalability and thus enable a trade-off between recognition performance and bit-rate. A simple approach to achieve scalability is to transmit only a subset of the 12 MFCCs. Thus some coefficients are "dropped" at the encoder, and set to zero at the decoder so that the recognition algorithm need not be modified, even if the number of coefficients transmitted varies over time. The relative importance of each MFCC can be determined experimentally by observing the degradation in recognition performance over a large training set when each coefficient is dropped. The coefficient that results in the largest drop in recognition performance when being set to zero is thus deemed the most important coefficient. For the TI46-Word digit database and the TI46-Word alphabet database, the coefficients were ordered, from least important to most important, as 11,10,9,6,8,4,5,7,1,3,2,0 (i.e., 11 would be dropped first, then 10 and so on) and (10,11,8,9,6,7,4,5,1,2,3,0), respectively. This approach is denoted "ad hoc pruning" in our experiments.

Since the speech recognizer has been trained with a full set of coefficients, ad hoc pruning may result in significant loss in recognition, especially when many coefficients are dropped for each frame. In addition, improving the performance of a pruning technique may require to select a different subset of coefficients to be pruned for each frame, which would then require overhead information to be sent to the decoder. As a simple alternative we use USQ with a dead zone (midthread quantizer) to quantize the prediction errors for all coefficients. Scalability is possible by changing the quantization step size: coarser step size results in lower rate and vice versa, while in the ECSQ case a completely different quantizer would have to be designed for each rate. Moreover, whenever a prediction error is quantized to zero we use the predicted value for that coefficient (rather than setting the coefficient to zero), which tends to affect less the recognition performance than pruning the coefficient altogether. Thus, USQ offers the advantages of low complexity encoding, simple design (no training is required) and easy scalability.

It should be noted that different frames can contain a different number of non zero coefficients based on the values of the prediction errors. Since the prediction errors are scaled by the pre-computed standard deviations and the higher MFC prediction errors have larger standard deviations, it is more likely that the prediction errors for the higher MFCCs be quantized to zero. This is desirable because, as mentioned above, the lower MFCCs tend to be more important for recognition. Because we use a Huffman coder to entropy encode the USQ indices, the lowest bitrate achievable is 1 kbps (the minimum is 1bit/coefficient, which corresponds to 12bits/frame, with one MFCC

frame being computed every 12 ms). To achieve lower bitrates, a bitmap can be transmitted to the decoder to indicate the position of the non-zero coefficients in every frame, and the non-zero coefficients can be entropy coded by the Huffman coder. The bitmap can be efficiently encoded using run length coding.

3 Experimental results

3.1 Experimental conditions

An HMM based recognizer (HTK2.1) was used to test the MFCC encoders developed in Section 2. The speech utterance was segmented using overlapping Hamming window of length 24 ms, with adjacent windows separated by 12 ms. 12 MFCCs derived from each segment of the speech utterance were used as the front-end. A left to right HMM with 4 states and 2 Gaussian mixtures was trained with unquantized MFCC front-end data, for each utterance. Diagonal covariance matrices were used. The training was done using two states of maximum likelihood (ML) and expectation maximum (EM). The baseline performance was determined by recognizing speech with unquantized MFCCs. The MFCC encoders proposed were tested with recognition experiments using encoded MFCCs. As comparison, experiments were performed on speech which had been coded by two low-bit speech encoders MELP (2.4 kbps) and FS-10 (CELP, 4.8 kbps). In this case, MFCCs were computed from the decoded speech. Better recognition performance may be possible by using waveform based speech coders (PCM, ADPCM), but these would require much higher bitrates (64 kbps, 32 kbps). So these high rate speech coders were not considered in our evaluation. Comparison was also done using methods similar to those reported in [3] and [4].

Recognition experiments were done for two databases, the TI46-Word digit database, which contains discrete utterances of digits and the TI46-Word alphabet database, which consists of discrete utterances of letters. The MFCC encoders proposed were designed based on the front-end data from the digit database and the same encoders were used for both databases. The HMM training was done with 80 utterances from 4 male and 4 female speakers (80 utterances from 8 male speakers) and the total number of test utterances used was 3320 (1260) for the alphabet (digit) experiment. Test utterances were from the same speakers of the training data, but different utterances.

3.2 Discussion and Conclusions

From Figures 1 and 2, it can be observed that the USQ technique, achieves recognition performance of 80.18% and 98.74% at bitrates of 0.95 kbps and 1.02 kbps for the alphabet and digit databases respectively. The respective recognition performances with unquantized MFCCs were 82.86% and 99.79%. This shows that while the degradation in recognition performance by encoding the MFCCs is small there are substantial savings in the bitrate. The advantage obtained by encoding the MFCCs as opposed to encoding speech and extracting the MFCCs from the decoded speech can be seen by using standard low bitrate speech coders for the same databases. A MELP speech encoder at 2.4 kbps achieved 75.15% and 98.85% recognition performance and a FS-10(CELP) speech coder at 4.8 kbps achieved 74.31% and 95.25% recognition performance for the alphabet and digit database respectively. It is clear that the recognition-rate performance of the MFCC encoders proposed is better than recognition-rate performance of the speech coders considered. This gain in recognition performance and reduction in bitrate required is not surprising because the speech coders have been optimized to preserve the perceptual quality of the speech, while the MFCC encoders are designed to maximize recognition performance.

Comparing the results obtained with previously proposed techniques, it can be observed from Figures 1, 2 and 3 that the methods proposed in this paper outperformed an algorithm similar to that reported in [3]. While the recognition performance of a VQ based technique, similar to that proposed in [4], is good, this method has the disadvantages of higher bitrate and higher encoding complexity, when compared with the methods proposed in this paper.

It is also evident from Figures 1 and 2 that the combined recognition-rate performance of USQ is better than the recognition-rate performance of ECSQ. This indicates that implicit pruning (by increasing the step size of the quantizers) of the MFCCs gives better results when compared to ad hoc pruning. The scalability of the proposed methods can be seen from Figures 1 and 2, which show the trade off in bitrate and recognition performance. By accepting lower recognition performance, we can operate at a lower bitrate (with more bits available for channel coding); this feature will be useful in situations where the channel conditions are varying such as in wireless communications.

The reduction in complexity at the client side by using our method can be seen from Table 1. Running the speech recognizer locally would require significant more computation and memory than quantization. For example recognition requires almost a factor of 3 more computation time than USQ.

The experiments on coefficient pruning offer several

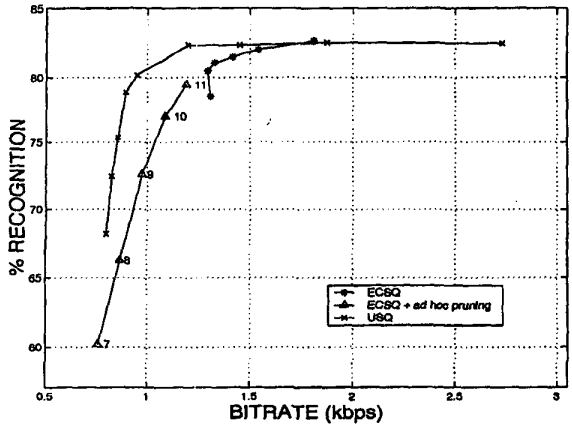

Figure 1: Recognition performance of the different encoders for alphabet database. Numbers next to the points indicate the number of coefficients retained. The scalability of the encoders can be seen by the Bitrate/Recognition performance tradeoff. Recognition performance with MELP was 75.15% and with FS-10 was 74.31%. The recognition performance with unquantized MFCCs was 82.66%.

Speech recognition	ECSQ	USQ
0.156 s	0.067 s	0.047 s

Table 1: Cpu time (in seconds, on a sun workstation) required to recognize a utterance from the digit database, and time required to encode it. The times shown for ECSQ and USQ also include the time for entropy coding, as well as the time required to compute the MFCCs (also included in speech recognition). The encoding and recognition times can be expected to be much higher for a portable device.

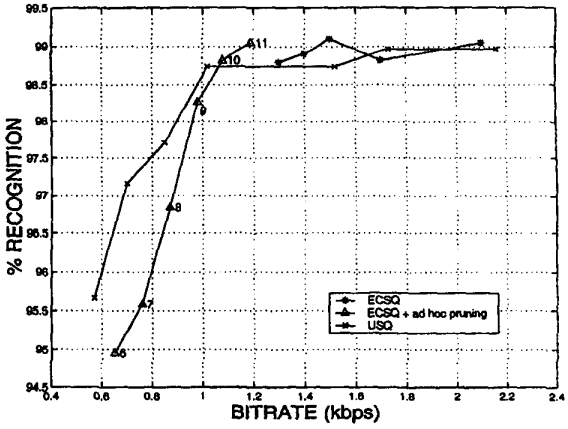

Figure 2: Same information as in Figure 1 except that the database is the digit database. Recognition performance with MELP was 98.85% and with FS-10 was 95.25%. The recognition performance with unquantized MFCCs was 99.79%.

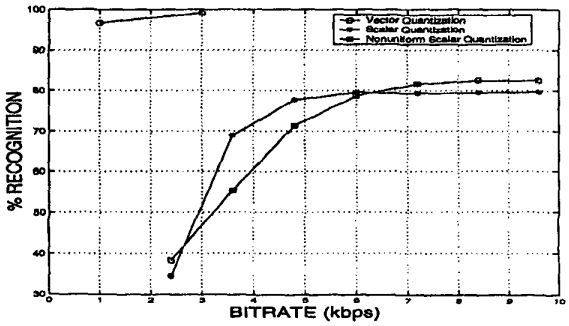

Figure 3: Recognition results with techniques based on scalar quantization[3] on the alphabet database and linear prediction and vector quantization[4] on the digit database.

interesting conclusions. It is observed that all MFCCs are not equally important for recognition. Also the importance of the coefficients varies from frame to frame. A reasonable conclusion that can be drawn from the results is that lower MFCCs are more important than the higher MFCCs. A clearer understanding of the effect of quantization and coefficient pruning on recognition will enable development of optimal solutions for joint compression and recognition of speech.

References

[1] S. Bayer, "Embedding speech in web interfaces," in *Proc ICSLP*, (Philadelphia, PA), pp. 1684–1688, October 1996.

[2] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 4, pp. 357–366, 1980.

[3] V. V. Digalakis and L. G. Neumeyer, "Quantization of cepstral parameters for speech recognition over the world wide web," *IEEE Journal on Selected Areas in Communication*, vol. 17, pp. 82–90, January 1999.

[4] G. N. Ramaswamy and P. S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in *IEEE ICASSP 1998*, pp. 977–980, 1998.

[5] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy constrained vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 31–42, January 1989.