PRODUCT-CODE VECTOR QUANTIZATION OF CEPSTRAL PARAMETERS FOR SPEECH RECOGNITION OVER THE WWW

V. Digalakis[1,2]*, L. Neumeyer*[2] *and M. Perakakis*[1]

(1) Dept. of Electronics and Computer Engineering
Technical University of Crete
Hania, 73100, GREECE

(2) SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, USA

ABSTRACT

We follow the paradigm that we have previously introduced for the encoding of the recognizer parameters in a client-server model used for recognition over wireless networks and the WWW, trying to maximize recognition performance instead of perceptual reproduction. We present a new encoding scheme for the mel frequency-warped cepstral parameters (MFCCs) that uses product-code vector quantization, and we find that the required bit rate to achieve the recognition performance of high-quality unquantized speech is just 2000 bits per second. We also investigate the effect of additive noise on the recognition performance when quantized features are used, and we find that a small increase in the bit rate can provide the necessary robustness.

1. INTRODUCTION

Speech-enabled applications over wireless networks and the World Wide Web (WWW) are recently attracting more and more attention [1-4]. In [2] we introduced a client-server speech recognition system for communication channels with limited bandwidth, a realistic assumption in applications that communicate over the Internet or through wireless channels. Speech captured by the clients may be deployed on heterogeneous environments, such as personal computers, smart devices, and mobile devices. The front-end processing takes place locally at the client, and the information is sent to the server. The server recognizes the speech according to an application framework and sends the result string or action back to the client.

In [2] we focused on the encoding of the speech-recognizer features, and we showed that the data compression problem for state-of-the-art hidden Markov model (HMM) based speech recognition systems differs from the traditional speech-coding problem: the optimization criterion is recognition accuracy instead of perceptual quality of the reproduced data. We decided to encode the mel frequency-warped cepstral parameters (MFCCs) that are typically used in most state-of-the-art speech recognition systems. The encoding scheme consisted of simple scalar nonuniform quantization of the MFCCs, and we showed that the error rate of a high-quality (full bandwidth) speech recognizer could be maintained by encoding at 3 kbps. In this paper, we improve the encoding scheme by quantizing subvectors, rather than the scalar parameters themselves. We show that a significant reduction in bit rate can be achieved in this way. In addition, we investigate the sensitivity of the proposed encoding schemes in an additive noise environment.

2. CLIENT-SERVER MODEL

In a client-server model, we can run the front-end processing (the feature extraction) at the client side on a wide range of machines, since feature extraction is only a small part of the computation in a speech recognizer. The representation of the speech signal used for recognition concentrates on the part of the signal that is related to the vocal-tract shape. Therefore, a significant reduction in bit rate over traditional speech coding schemes can be achieved if the recognizer parameters are encoded at the client and then transmitted to the server.

Of course, encoding and transmitting only the front-end processed tokens can become a disadvantage, since, without any representation of the speech associated with these tokens, the input speech cannot be labeled. As a result, it may not be possible to monitor in-service performance, or to collect labeled speech data for development and performance improvement. To overcome this limitation, and collect labeled data during the initial deployment of the application, it is possible to transmit the original speech encoded using a very-low-bit-rate coder as side information. This side information can be transmitted on top of the encoded front-end tokens during the development phase only.

The client-server model can be applied to the Internet, as well as to wireless channels. The Aurora Project is a joint initiative, where several companies, including Alcatel, Ericsson, IBM, Matra, Nokia, Nortel and Siemens, are working to establish a global standard for distributed speech recognition over wireless channels [3]. On the Internet, the client-server model has been adopted by the BBN SPIN (Speech on the Internet) system that was presented in [4]. Although the details of this system are not known, it was reported that it encodes speech at 3.8 kbps in a form suitable for discrete-density HMMs.

In our work, we follow the client-server model using the encoding scheme that is described in Section 3. We implemented a highly modular signal processing front end in Java to compute the MFCCs and encode the parameters. We verified that the system is fast enough to handle the feature extraction in real time, using a Pentium 166 MHz computer and a Java virtual machine (JVM) with a just-in-time (JIT) compiler. We also ran benchmarks to compare performance on the computation of the fast Fourier transform. We found that the optimized C code is twice as fast as the Java implementation. We believe that as the JVMs become more efficient the gap between C and Java performance will become even smaller.

The Java applet is downloaded from the server. By default, the Java security model prevents an applet from accessing native resources. There are various possible approaches to granting permission to access native resources. The various approaches for handling security policies in the Java model are beyond the scope of this paper.

3. PRODUCT-CODE VQ

For the client-server approach, we need only transmit the set of coefficients that will be used in recognition. Typical choices for the dimension of the feature vector and the rate at which it is computed are 13 and 100 times per second, respectively. Secondary features, like the first- and second-order derivatives of this feature vector that are also used in recognition, do not have to be coded and transmitted, since this information can be obtained at the server side. Hence, one needs only to quantize a total of 1300 parameters per second of speech. After the parameters are encoded at the client side and transmitted through the network, they are mapped to their centroids at the server and used as input to a continuous-density HMM recognizer.

In [2] we employed nonuniform scalar quantization of the MFCCs. Scalar quantization cannot take advantage of the relationships that may exist between different coefficients, and cannot assign fractions of a bit to a particular coefficient. To achieve these goals, one must to use subspace quantization of the cepstral coefficients. The MFCCs are partitioned into subvectors, and then the subvectors are encoded by using separate codebooks. The total number of codewords that represent the acoustic space is the product of the number of codewords used for the representation of each subvector. The same technique is also used for coding several types of speech analysis parameters, including log-area-ratios (LARs) in traditional speech coding applications [5].

In the general case, the dimensions of the subspaces used in the product code are larger than one. Although more complex variations of product codes exist, we are interested here in partitioned VQ, where we simply partition the cepstral vector into two or more nonoverlapping subvectors. Product codes provide significant savings in memory storage of the codewords and reduce the computational cost for separable distortion measures [5]. Both these types of savings are very important in our application, because of the large number of codewords that must be used for good recognition performance. Since the coding of the cepstral vectors takes place at the client, heavy memory and computational requirements can significantly limit the types of machines that can access a speech-enabled WWW site.

An important issue in the design of a product code is the method used to partition the feature vector into subvectors. A product code is optimal if the component vectors are independent and the distortion measure is separable [5]. Hence, one can partition the cepstral vector into subvectors by trying to satisfy the independence criterion. One approach is to partition the cepstral vector using the matrix of the estimated pairwise correlation coefficients of its elements. Each cepstral coefficient can be assigned to the subvector with the elements that are more correlated on average. An alternative, knowledge-based approach is to partition the vector of MFCCs into subvectors that contain consecutive coefficients, so that the most important low-order coefficients are grouped together.

Once the subvectors of the product code are formed, the next important design question is how to allocate the bits among the respective codebooks. Since we are interested in coding speech features for recognition, we have designed a bit-allocation algorithm that uses the word-error rate as a metric. Specifically, we start with an initial bit allocation to subvectors, and then increase the bit rate by adding bits to the subvectors that yield the maximal incremental increase in recognition performance as follows:

Initialization: Allocate the initial number of bits to subvectors and evaluate speech recognition performance. Set this as the current configuration.

Step 1: For each subvector, increase its allocated number of bits by one and evaluate speech recognition performance, keeping the number of bits assigned to each of the remaining subvectors as in the current configuration. Assign the additional bit to the subvector that resulted to the maximal increase in recognition performance, and set the new assignment as the current configuration.

Step 2: If the desired recognition performance has been achieved, or the maximum available bit rate has been reached, stop. Otherwise, go to step 1.

Any available metric can be used to evaluate speech recognition performance. In this work we have used the word-error rate (WER).

Although the above procedure is computationally expensive, because of the multiple recognition experiments that must be run at each step, it is executed only once during the initial design of the quantizer. If, however, a faster allocation scheme is desired, the total assigned bits in the second step can be incremented in steps of multiple bits.

4. EXPERIMENTS

To experiment with the quantization of cepstral parameters for speech recognition over the WWW, we have selected the air-travel information (ATIS) domain [6]. This was the domain of the first speech-enabled application over the WWW developed at SRI International [1].

The recognizer used throughout our experiments is SRI's DECIPHER^{TM} speech-recognition system [7]. It uses continuous-mixture density HMMs, with Gaussians that are shared across acoustically similar states. The signal processing consists of a filterbank-based front end that generated six feature streams: the

	Word-Error Rate (%)	
Bit Rate (bps)	Correlation-based partitioning	Knowledge-based partitioning
1400	18.77	11.71
1600	13.36	9.30
1800	10.24	8.10
1900	8.92	6.99
2000	8.38	6.63
2100	7.72	
2200	7.01	

Table 1: Bit rates and word-error rates for product-code VQ using five subvectors created by either a correlation-based or a knowledge-based approach.

Composition of subvectors by MFCC coefficients						
1,2	3,4	5,6,7	8,9, 10	11,12 ,13		
Number of bits assigned to each subvector at each iteration					Bit Rate (bps)	Word-Error Rate (%)
3	3	2	2	2	1200	16.79
5	3	2	2	2	1400	11.71
5	3	4	2	2	1600	9.30
5	3	4	4	2	1800	8.10
5	4	4	4	2	1900	6.99
5	5	4	4	2	2000	6.63

Table 2: Progression of the bit-allocation algorithm for the case of five subvectors. The bits assigned to each subvector, the total bit rate, and the corresponding word-error rate are shown at intermediate steps of the algorithm.

cepstrum, the cepstral energy, and their first- and second-order derivatives. Thirteen cepstral coefficients, including the cepstral energy, were used. The coefficients were computed at a rate of 100 times per second. A bigram language model was used throughout our experiments.

4.1 Baseline Performance

The performance of the baseline recognizer high-quality speech was evaluated at 6.55% WER using a test set of 34 male and female speakers with 400 utterances. In [2] we also measured the performance on telephone-quality speech, which is significantly lower and was measured at 12.7% WER. Compared with the telephone-quality baseline, the recognition performance did not degrade when the data was encoded using the G721 32-kbps ADPCM coding standard. However, when speech was encoded with the full-rate RPE-LTP GSM 13-kbps speech encoder used in cellular telephony, the WER increased to 14.5%. When the MFCCs were encoded at the client using nonuniform scalar quantization, we were able to duplicate the high-quality recognition performance of 6.55% WER at a bit rate of 3 kbps.

4.2 Product-Code Quantization Performance

Our experiments used product code VQ with a variable number of bits per subvector. In all our experiments, the codebooks for each subvector were estimated by running the generalized Lloyd algorithm on the same 800 utterances that were used to estimate the empirical distribution in the nonuniform scalar quantization experiments in [2]. The codebooks were initialized using binary splitting [5].

We first compared the two alternative approaches for partitioning the cepstral coefficients into subvectors. In Table 1 we present, for the case of five subvectors, the WERs of the correlation- and knowledge-based approaches at various bit rates, as we measured them at various stages of the bit-allocation algorithm. The five subvectors consisted of the cepstral coefficients {(1,5), (3,9,12,13), (4,6), (2,7,11), (8,10)} and {(1,2), (3,4), (5,6,7), (8,9,10), (11,12,13)} for the correlation-based and the knowledge-based partition schemes, respectively. We see that the knowledge-based partitioning exhibits significantly better performance at all bit rates, and converges to the unquantized WER of 6.55% at a lower bit rate than the correlation-based scheme. We found experimentally that the problem with the correlation-based partitioning was the very low correlation between the various cepstral coefficients, which resulted in somewhat arbitrary partitions. This problem can be resolved by measuring phone-specific correlation coefficients, rather than pooling all the speech data together. Given the exceptional performance of the knowledge-based partitioning, which achieved the WER of the unquantized speech at just 2000 bps, we adopted the knowledge-based scheme for the rest of our experiments.

We then examined the behavior of the bit-allocation algorithm for various numbers of subvectors in the product-code VQ. In Table 2 we present the case of five subvectors. The initial bit rate was 1200 bps, and the algorithm was initiated by distributing twelve bits to the five subvectors, as shown in the first row of Table 2. To speed up the process, the number of allocated bits was increased by a step of two bits in the first iterations of the algorithm (until 1800 bps), and by a single bit in the latter stages of the algorithm. We can see that the initial WER of 16.79% decreases very rapidly and approaches the unquantized-speech performance at 2000 bps. The significance of the low-order coefficients is also obvious: The additional bits are allocated to the low-order subvectors first, and the final bit allocation uses more bits for the first two subvectors, although they are composed of only two coefficients each.

In Figure 1, we have plotted speech recognition performance as a function of the bit rate for different numbers of subvectors in the product-code VQ (three and five), and for the nonuniform scalar quantization with a variable number of bits per coefficient. In the same figure, we also show the WER for nonuniform scalar quantization using two bits per coefficient.

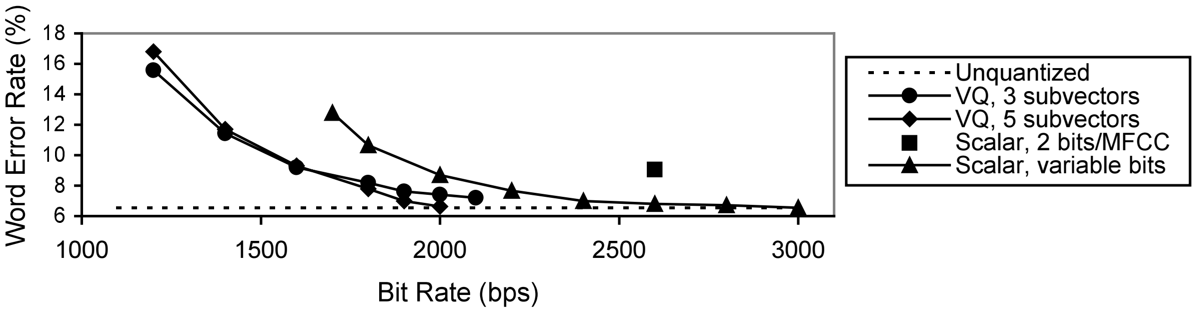

Figure 1: Recognition performance as a function of the bit rate for various types of MFCC encoding: nonuniform scalar quantization with constant and variable number of bits per coefficient; product-code vector quantization with different numbers of subvectors.

The partitioning of cepstral coefficients into subvectors for the case of five subvectors was given above, whereas for the case of three subvectors, the partitioning was {(1,2,3), (4,5,6,7,8), (9,10,11,12,13)}. Scalar quantization with a variable number of bits demonstrates significantly better performance than the scalar quantization scheme with a fixed number of bits per coefficient, reducing the WER to 6.81% from 9.04% at 2600 bps. Product code VQ, however, performs significantly better than scalar quantization at any bit rate. When comparing the three- and five-subvector cases, we see that they behave similarly for low bit rates (below 1800 bps), but then the five-subvector scheme converges faster to the unquantized speech performance.

In a final set of experiments we evaluated the robustness in the recognition performance when product-code encoding of MFCCs is used. The test set was distorted with additive noise at a signal-to-noise ratio of 24 dB. The noise was recorded in a moving vehicle, and is the same noise that was used in the WSJ HUBX evaluation. In Table 3, we can see that the performance of the baseline system without encoding of the MFCCs degraded to 8.51% from 6.55% WER at the SNR of 24dB. The performance of the 2 kbps quantized system degraded significantly more, dropping to 12.19% WER. The situation did not improve when the VQ codebooks were retrained under matched train conditions, that is, with 24 dB additive noise, and the WER improved only slightly to 11.89%. We then applied the bit allocation algorithm in the 24 dB noisy test, initializing it with the 2 kbps configuration. In Table 3 we can see that 27 bits are now required to achieve the unquantized performance in the 24 dB condition, corresponding to a bit rate of 2.7 kbps.

5. ACKNOWLEDGMENTS

This work was accomplished under contract to Telia Research of Sweden and by SRI research and development funds.

6. REFERENCES

1. L. Julia, A. Cheyer, L. Neumeyer, J. Dowding and M. Charafeddine, "http://www.speech.sri.com/demos/atis.html," *Proceedings AAAI'97,* Stanford, CA, March 1997.

2. V. Digalakis, L. Neumeyer and M. Perakakis, "Quantization of Cepstral Parameters for Speech Recognition over the WWW," *Proceedings ICASSP'98,* Seattle, WA, May 1998.

3. The Aurora Project, announced at Telecom 95, "http://gold.ity.int/TELECOM/wt95", Geneva, October 1995.

4. D. Stallard, "The BBN SPIN System", presented at the Voice on the Net Conference, Boston, MA, Sep. 1997.

5. J. Makhoul, S. Roucos and H. Gish, "Vector Quantization in Speech Coding," *Proceedings of the IEEE,* Vol. 73, No. 11, pp. 1551-1588, November 1985.

6. P. Price. "Evaluation of Spoken Language Systems: The ATIS Domain,*" Proceedings of the Third DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June 1990, Morgan Kaufmann.

7. V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," *IEEE Trans. Speech Audio,* pp. 281-289, July 1996

Test SNR (dB)	Train-VQ SNR (dB)	Encoding bits/subvector	Word-Error Rate (%)
Clean	No encoding	-	6.55
Clean	Clean	5 5 4 4 2	6.63
24	No encoding	No encoding	8.51
24	Clean	5 5 4 4 2	12.19
24	24	5 5 4 4 2	11.89
24	Clean	5 5 5 4 2	11.18
24	Clean	6 5 5 4 2	10.49
24	Clean	6 6 5 4 3	9.47
24	Clean	7 6 5 4 4	9.32
24	Clean	7 6 5 4 5	8.94

Table 3: Quantization in noisy conditions