HOW GOOD IS YOUR β ? - OBSERVATIONS ON VQ TRAINING RATIOS

John S. Collura, Thomas E. Tremain
9800 Savage Rd. Ft. Meade MD 20755-6000
E-mail: jscollu@alpha.ncsc.mil

ABSTRACT

A growing number of state of the art speech coding algorithms use vector quantization (VQ) to quantize spectrum information. VQ code books are created from a set of training vectors which are drawn from and representative of the overall data being quantized. These training vectors are partitioned into a set of clusters whose centroids represent the region of the partition and are called code vectors. Of specific interest to this paper is the ratio, β, of the number of training vectors to the number of code vectors [1]. The goal of this paper is to provide guidance on appropriate levels of training data regardless of code book size. Of particular significance is the empirical determination of a minimum β value of 128 training vectors per code vector for full vector code books.

INTRODUCTION

Low rate speech coders must transmit a representation of the speech spectrum to a receiver for reconstruction of the speech signal. For simplicity, this paper will restrict the representation of the spectrum parameters to that of the line spectrum frequency (LSF) parameters discussed in [2].

A literature search by the authors did not produce substantive guidance about the quantities of training data required to create VQ code books, except "include as much data as possible". This paper seeks to provide guidance for the basic question "How much training data is enough to represent the source to be quantized?".

This paper will provide a brief introduction to what a vector quantizer is and does, followed by a discussion about training databases. Experimental data will then be presented and empirical guidelines for training ratios will be introduced. Finally, based upon these guidelines, a discussion and conclusions will be presented.

VECTOR QUANTIZATION

Vector quantization is a process where the elements of a vector are jointly quantized. Vector quantization is more efficient than scalar quantization by accounting for nonlinear dependencies and vector dimension as well as linear dependencies and the shape of the probability density function [3].

Linear predictive filter parameters in the form of line spectral frequencies collectively form the vectors of interest in this work. For simplicity, we refer to these vectors as $\mathbf{x} = [x_1, x_2, \ldots x_p]^T$ for a pth order LP filter. The quantized version of the vectors is represented by the symbol **y**, where $\mathbf{y} = \mathbf{q}(\mathbf{x})$ or **y** is the quantized value of **x**. Vector quantization is accomplished through a mapping of the continuous input parameter vector **x**, into one of a set of discrete vectors **y** called code vectors. These code vectors are preselected through a clustering or training process to represent the training data, and stored in a table called a code book. Generic vector quantization is performed by comparing the input vector **x** to each of the code vectors $\mathbf{y}_i$ and selecting the code vector which achieves minimal difference.

$$D(\mathbf{x}, \mathbf{y}_i) \le D(\mathbf{x}, \mathbf{y}_k) \quad \text{for all } k, k \neq i$$

The index which is assigned to the selected code vector is then transmitted to the receiver for reconstruction. The preferred measurement for the calculation of the difference is the log spectral error measurement [4]. Due to computational considerations most vector quantizers use either the squared Euclidean distance or the weighted squared Euclidean distance measurement when searching the code books [1].

For a given speech coder, there are identical copies of the code book located in both the transmitter and the receiver. The transmitter identifies the code vector with the minimum distance to the input vector and transmits the index or address of the code vector to the synthesizer. The synthesizer then simply performs a table lookup to obtain a quantized copy of the input vector.

There are many different kinds of vector quantizers available to designers of speech coding algorithms [1][3][5]. This paper will narrow the focus onto just two kinds, the full vector and the split vector quantizers. The full vector quantizers simply quantize the spectrum as described above, while the split vector quantizers divide the vector into a set of 2 or more sub-vectors. Each of these sub-vectors is then independently quantized subject to certain constraints [4].

DATABASES

Regardless of the training procedure used to generate the code books, there are two issues which are of paramount importance. These are the statistical significance of the training set and the appropriate representation of the anticipated source to be quantized.

The size and makeup of speech databases are of critical importance to the training process in vector quantiza-

0-7803-2431-5/95 $4.00 © 1995 IEEE

tion. The number of training vectors in the database has a direct effect on the type of structure a given algorithm might produce. This decision is based on the ratio of available training vectors to desired code vectors. If there are too few training vectors, and a large code book is required, then it may be advantageous to use a multistage or split structure in which each step requires fewer code vectors. The rationale is as this ratio becomes too small, the resulting code books will show signs of undertraining when tested on disjoint speech databases. Undertraining becomes apparent when the test data distortion exhibits an uncharacteristically large distortion measurement. Testing on a disjoint data set is done to maintain the validity of the test. If possible, these disjoint vectors should not be drawn from the same overall training set conditions. The only caution here is that the test data should not represent unrealistic conditions which have no probability of occurring in operational environments.

Vector quantization training procedures require a rich combination of source material to produce code books which are sufficiently robust for quantization of data not represented in the training set. Examples of some of the conditions which might enrich the training set include varying microphones, acoustic background environments, languages and gender. The goal of collecting databases is to obtain as large and diverse a set of vectors as possible in order to represent a reasonable approximation to the expected input data being quantized. This goal is very difficult to reach as there are no guarantees that new or unforeseen applications may not arise, or that the selected database might be biased in some way or other, e.g. too much silence. In general, a large diverse training set is required to produce a reasonably robust code book while at the same time providing a statistically significant basis for the mathematical models used to create the code books used in quantization.

EXPERIMENTS

Two basic experiments were performed to attempt to determine a reasonable value for β, where β is defined as the ratio of the number of training vectors N to the number of code vectors M, $M=2^L$ for an L bit code book.

The first experiment was to fix the code book size at $M=2^L=1024$ code words and train a set of code books with varying quantities of data. The smallest quantity of data was simply the code book size of 1024 vectors or a β ratio of 1. Each successive code book was generated by doubling the amount of training data of the current code book. So a code book generated using 16384 training vectors would have a β ratio of 16. Once a set of code books where β varied from 1 to 1024 were created, 2 tests were performed on each code book.

The log spectral distortion measured in the range of 100 Hz to 3 KHz was used to determine the distortion of each of the following tests. The first test measured the average log spectral distortion when quantizing the training set used to create that particular code book. The second test measured the average log spectral distortion when quantizing a disjoint set of 100,000 vectors. These vectors were not only disjoint from the training vectors, but were drawn from a completely different database. In this instance the testing database was the TIMIT test database for dialect regions 1 through 5. The results obtained from this experiment are shown in Figure 1.

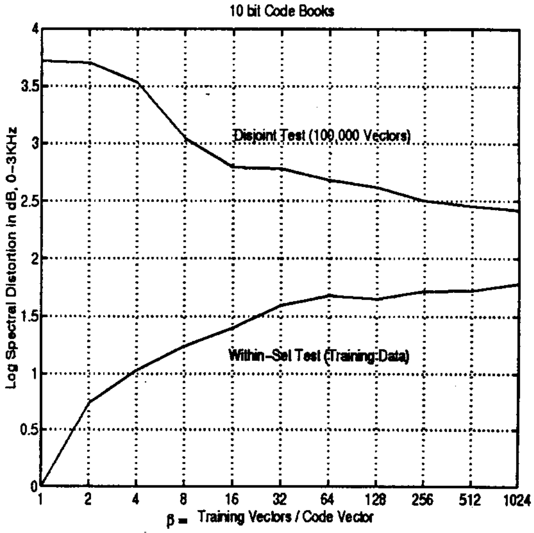

Figure 1: Within-set vs. Disjoint testing

The second experiment varies the training ratio β by varying the number of code vectors rather than the number of training vectors. For this experiment each code book was trained on the identical database of $M=2^{20}$ vectors, and as before each code book has a different β value. A major difference in this experiment is that split-2 and split-3 structures were used in addition to the full vector structure. The same 100,000 vector disjoint test set described in experiment 1 was used for testing in this experiment. Code books were searched using the weighted squared Euclidean distance criteria. The average log spectral distortion was then computed between each quantized vector and the original test vector. The results for this experiment are shown in Figure 2, which also displays the average log spectral distortion for the 34 bit Federal Standard 1016 CELP quantizer [6]. In Figure 2, β must be calculated from the data in the graph. This is done by taking the training set size of $M=2^{20}$ or L=20 bits and subtracting the code book size to get the number of bits in β. If a split vq code book is being considered, subtract the largest of the code book sizes. To derive the actual ratio, raise the number of bits to the power of 2, i.e. if the code book size is $L_c=13$ bits, and the training set size is $N=2^{20}$, then the ratio is $N/M = 2^{20}/2^{13} = 2^7$, or β=128. In the cases where there are split code books, higher performance was measured when the code book sizes were approximately the same. In the cases where this was not possible, the extra bit was assigned to the lower code book. A 24 bit split-2 VQ would have a 12 bit lower code book and a 12 bit upper code book, while a 25 bit split-2 VQ would have a 13 bit lower code book and a 12 bit upper code book.

DISCUSSION

The problem of how to empirically define an appropriate value for β is the central issue of this paper. The experiments described above attempt to provide insight into this problem. The two experiments do not comprise as persuasive an argument independently as when they are considered together.

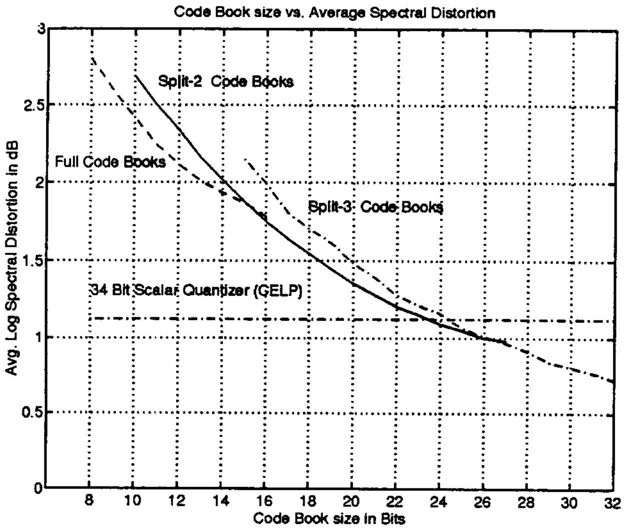

Figure 2: Code Book size vs. Distortion.

When Figure 1 and Figure 2 are jointly considered, Figure 1 indicates that a minimum β of around 64 to 128 training vectors per code vector are required to train a full vector code book. This is based upon an interpretation of the data in the graphs. As the value of β is increased in Figure 1, the within-set measurement increases in distortion, and the disjoint measurement decreases in distortion. The point where β is below about 64 experiences a dramatic increase in distortion on the disjoint trace. Figure 1 also demonstrates the need for measuring the distortion of vector quantizing code books on disjoint data sets. To state the obvious, no matter how much data is used to train the vector quantizing code books, when you test on the training data, the results will always appear better than what can be achieved in an operational system. Testing for Figure 1 was conducted using two distinct measures, the first is the squared Euclidean distance between parameter vectors used to search the code books, and the second is the average log spectral distortion used to measure the actual distortion as previously mentioned. The slightly non-monotonic behavior of the plots can be attributed to this difference. If one were to search the code books using the log spectral distortion between the input vector and the code vectors, then the traces in figure 1 should be monotonic.

When the data from figure 2 is considered, a clearer picture of the proper β size emerges. Figure 2 graphs code book size versus distortion for full, split-2 and split-3 code books. Under ideal circumstances where each code book is trained on adequate quantities of data, the traces for each code book type should never cross [5]. The trace for the full vector code books should always measure lower distortion than either split-2 or split-3 code books for equivalent code book sizes. This is also true when comparing split-2 against split-3 code books, and so on. Figure 2 points out that when the ratio of training vectors to code vectors, β, becomes too small, the code books become less efficient. This condition is known as undertraining, and signs of undertraining become apparent when the slope of the traces flatten out, rather than where they cross. Code books begin to show signs of undertraining as β drops below 128 training vectors per code vector. In Figure 2, observe the range of 13 to 16 bit full code books, and the range between 23 and 27 bit split-2 code books. For the full code book case, as the ratio of the number of training vectors to code vectors falls below β=128, the distortion begins to level off to the point where a split-2 code book of the same size actually outperforms the full code book. When the split-2 ratio drops to about β=256 or so, the split-3 code books begin to outperform the split-2 code books. Observe also that this phenomenon occurs at higher thresholds (β=256) for the split-2 code books than for the full code books. This is due to the splitting process destroying the inter vector dependencies. It should be noted that the determination of an appropriate β size for split-3 code books is outside the scope of this paper. This is due to the lack of data for code books split into greater than 3 parts.

Table 1 provides a detailed look at the test results for the regions of Figure 2 where the full code book trace crosses that of the split-2 code book and where the split-2 trace crosses that of the split-3. For comparison, an additional data point will be included as a baseline for each crossing, that is, a point where both code books are equally well trained. Code book efficiency is defined as the percentage of code words selected from a code book when quantizing the entire test database. If a 12 bit full code book were to use 3821 of the available 4096 code words, then this code book is 93.29% efficient. One needs to exercise a bit of caution when looking at very low efficiency ratings for the larger code books due to the relatively small ratio of test vectors to code vectors. The average log spectral distortion and the percentage of frames whose log spectral distortion falls within 1 dB bins from 0 to 4 dB and the percentage with measurements greater than 4 dB are also presented. The minimum value for β was also included in Table 1 to allow for easier comparisons with the graphs. B in Table 1 refers to the minimum value that can be determined for that code book, i.e. if one had a 27 bit split-2 code book with a 14/13 split, then β would be $2^{20}\text{-}2^{14} = 2^6 = 64$. It should be noted here that outlier frames whose distortion is greater than 4 dB comprise a larger percentage of overall frames than that called for in the literature [4]. This can be accounted for by reiterating that the test conditions are completely disjoint from the training conditions, and that the amount of test data exceeds what has been reported in the literature by 1 or more orders of magnitude.

The first two rows in Table 1 demonstrate that a 12 bit full code book with a $\beta > 128$ outperforms a 12 bit split-2 code book whose component code books each satisfy this condition. While the efficiency of each of the 12 bit code books is comparable, the full code book displays better performance with respect to outliers. This same comparison applies between the 18 bit split-2 and it's 18 bit split-3 counterpart with the split-2 code book outperforming the split-3 by a respectable margin. Next we compare the performance of the 16 bit full code book where β=16 to the 16 bit split-2 code book where β=4096. Intuition tells us that full code books should outperform split code books, however this is not the case. There is no hard and fast rule for accounting for the decreased efficiency experienced with the larger full code book. The efficiency of the 16 bit full code book is a factor of 3 smaller than that of the split-2 code book counterpart. The average log spectral distortion is virtually identical for both code books, as is the outlier performance. This difference in code book efficiencies cannot simply be the result of a smaller code word to test vector ratio, and must be the result of some other phenomena such as undertraining.

The same comparison can be applied between the split-2 code books and the split-3 code books. This time 18 bit code books are used as a baseline. It is apparent upon looking at the 18 bit data in Table 1 that the split-2 code book with β=2,048 outperforms the split-3 code book with β=16,384 in every category, and that the efficiencies of the 2 code book structures are comparable. The data for the 27 bit code books paints a different picture. The split-3 code book structure where β =2,048 for each code book is more efficient than the split-2 code book where β=64 for the lower code book and β=128 for the upper code book. Here, the average log spectral distortion for the two code books is virtually identical with the number of outliers reduced by 50% for the split-3 structure.

CONCLUSIONS

The problem of inadequately sized databases for training vector quantizing code books was introduced. Based upon two related experiments, empirical guidelines for the size of the training set ratio β have been established and presented for both full vector and split vector code books. These ratios are β=128 and β=256 for the full and split-2 vector code books respectively. These values represent the minimum ratios that designers should use for creating properly represented code books. For code books which have better outlier performance, greater quantities of training data are recommended.

REFERENCES

[1] A. Gersho, R.M. Gray "Vector Quantization and Signal Compression", Kluwer Academic, 1992

[2] F.K. Soong and B.H. Juang, "Optimal quantization of LSP Parameters", IEEE Transactions on Speech and Audio Processing, Jan 1993

[3] J. Makhoul, S. Roucos, H. Gish, "Vector Quantization in Speech Coding", Proceedings of the IEEE, November 1985

[4] K.K. Paliwal and B.S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 bits/frame", IEEE Transactions on Speech and Audio Processing, Jan 1993

[5] J.S. Collura, "Vector Quantization of Linear Predictive Coefficients", In *R.P. Ramachandran, R.J. Mammone editors, "Digital Speech Processing: Theory, Applications and Techniques", to be published. Kluwer Academic Publishers, 1994*

[6] United States Federal Standard 1016, "Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bps Code Excited Linear Prediction (CELP)", February 1991

Code Book Size/Type	Min β	Code Book Efficiency	Avg. LSD	0 to 1db	1 to 2db	2 to 3db	3 to 4db	>4db
12 bit Full	256	93.29%	2.116366	2.40%	46.85%	39.88%	8.94%	1.93%
12 bit Split-2 6/6	16,384	100% / 100%	2.328936	0.99%	37.06%	45.86%	12.26%	3.83%
16 bit Full	16	30.71% *	1.782853	7.83%	61.49%	25.94%	3.86%	0.88%
16 bit Split-2 8/8	4096	99.2% / 100%	1.751705	6.79%	67.42%	22.50%	2.90%	1.09%
18 bit Split-2 9/9	2048	98.44% / 100%	1.545972	13.04%	71.25%	13.11%	1.80%	0.81%
18 bit Split-3 6/6/6	16,384	100% / 100%/ 100%	1.731116	7.42%	67.40%	20.20%	3.55%	1.44%
27 bit Split-2 14/13	64	66.67% / 95.58%	0.967924	64.13%	33.06%	2.00%	0.50%	0.30%
27 bit Split-3 9/9/9	2048	98.43% / 100%/ 100%	0.963513	64.90%	32.49%	3.02%	0.44%	0.15%

Table 1: Snapshot of Experiment 2 (Figure 2) Test Results
* artificially low due to the small ratio of code words to test vectors.