Training Sequence Size and Vector Quantizer Performance

Pamela C. Cosman Keren O. Perlmutter Sharon M. Perlmutter
Richard A. Olshen* and Robert M. Gray

Information Systems Laboratory
Durand Building
Stanford University
Stanford, CA 94305-4055

*Division of Biostatistics
Health Research and Policy Building
Stanford University
Stanford, CA 94305-5092

Abstract

We examine vector quantizer performance as a function of training sequence size for both tree-structured and full-search vector quantizers. The performance is measured by the mean-squared error between the input image and the quantizer output at a given bit rate. The training sequence size is measured either by the number of training images, or by the number of training vectors. When the training vectors are counted, they are selected randomly from among the training images. For every training sequence size, vector quantizers are developed from several different training sequences and the distortion is calculated for different test sequences in a cross validation procedure. Preliminary results suggest that plots of distortion vs. number of training images (n) follow an algebraic decay of the form $An^{-\alpha} + B$ as expected from analogous results of learning theory.

1 Introduction

Vector quantizers are generally trained on a sequence of images. Insufficient training data causes the vector quantizer to become "over-trained," in that it becomes very good at coding the training vectors it was supplied with, but is likely to be highly unreliable when coding anything else. Too much training data, on the other hand, should not degrade VQ performance, but can slow down research and development efforts. The amount of training data needed to obtain reliably a preassigned level of performance from the vector quantizer has not been studied and researchers employ "rules of thumb" to decide how much training data to use. The purpose of this work was to address this question, and generally to study the form of the relationship between performance and training sequence size. The study of training sequence size for vector quantization using learning theory is independently being pursued by David Cohn at the University of Washington. Separate training and test images were chosen from a set of 12 mid-sagittal MRI brain scans of different individuals. The two vector quantizers considered were the unbalanced tree-structured quantizer (TSVQ) and the full-search quantizer (FVQ). The quantizers were developed using the Generalized Lloyd Algorithm, and the TSVQs were grown one node at a time in a "greedy" fashion as described in [5, 4, 3] and then pruned back using the generalized Breiman, Friedman, Olshen and Stone (BFOS) algorithm [1]. In all cases the vector dimension was 4 (2×2 pixel block). The performance was measured by the mean-squared error (mse) between the input image and the quantizer output at a given rate.

2 Approach

We used two sampling approaches in this study. First, images were considered as sampling units, which implies that with a total of 12 images available, one would choose n as the actual training images. This is the approach generally used in practice. Second, vectors were considered as sampling units, which implies that we would randomly choose n images' worth of training vectors from all 12 images, rather than a specific set of n images, to create a training sequence of size n. With the first approach, including a highly unusual image as one of the training images could result in high variability. With the second approach, we take only some of the vectors from the unusual image for all training sequence sizes, thus lessening the effect of a very unusual image. On the other hand, if the images all have very similar pixel intensity distributions,

1058-6393/91 $01.00 © 1991 IEEE

then one specific image might be more representative of the distribution of another image than a random sampling of 5 images would be. It is thus not obvious *a priori* which sampling method will yield better or more trustworthy results.

2.1 Images as Sampling Units

Training vectors were obtained from a subset n of the $I = 12$ images, where $n \in \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. A training sequence from less than 1 image was obtained by randomly choosing a subset of the training vectors from 1 image. Four of the 12 images were used separately as the test images, and the results for the 4 images at a specific size n were then averaged to obtain each data point on the distortion vs. training sequence size plot.

In order to avoid biased results, images used as a source for training vectors were not used as a source for the corresponding test vectors. In addition, we used a split sample scheme, similar to cross-validation, where we took more than one subset of size n as a source for training vectors whenever possible, and then averaged over the results obtained for the different subsets. The number of different training sequences used to obtain each data point varied slightly due to the limited number of images available, and because we wanted to ensure that particular training vectors never composed more than one training sequence for a specific training sequence size n. Four different training sequences were used separately to obtain results for the training sequence sizes of $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, 1 and 2. Two different training sequences were used separately to obtain results for training sequence sizes 3 and 4. Only one training sequence was used to obtain results for the sizes 5, 6, 7, 8, 9, and 10.

2.2 Vectors as Sampling Units

All training vectors from all 12 images were randomly reassigned to one of 12 new fictitious "images." For an m-fold cross-validation for $n < I$, we pick

$$t = v[\frac{m}{m-1}][\frac{n}{I}] \tag{1}$$

vectors from each new "image." n denotes the number of training images, v denotes the number of vectors in each "image," and I denotes the total number of images available. In our study, we used $m = 10$-fold cross-validation, with $v = 16384$ vectors/image, and $I = 12$. Nine-tenths of these t vectors are obtained from each new "image," and these are combined together to form the training sequence. The remaining one-tenth of the t vectors from each new "image" are combined to form the test sequence. Ten different training sequences of size n and ten corresponding different test sequences are then produced by choosing different nine-tenth and one-tenth combinations. For example, to investigate the distortion when one training image's worth of vectors is used, i.e., $n = 1$, we picked 16384(10/9)(1/12) or approximately 1520 training vectors from each image (rounding ensured an equal number of training vectors from each image).

2.2.1 Bias-Variance Tradeoff of Nesting

Training sequences of different sizes were nested. For example, half of the training vectors that form a training sequence of size 2 are those vectors that composed the training sequence of size 1.

The decision to nest the increasing sets of training vectors or not is a choice of trading off bias and variance that is familiar in the statistical estimation of functions and also in signal processing. The tradeoff that we describe is a very subtle analogue of the problem that arises when one filters a periodogram to estimate a spectral density. An identity filter which simply passes the signal without change, i.e., a filter with a delta function as its pulse response and a constant transfer function, has little bias but a variance that does not tend to 0 with increasing sample size. On the other hand, a long comb filter for smoothing the input signal, i.e., a filter with a very narrow bandwidth, will have little variability, but will have a persistent bias that does not tend to 0 with increasing sample size. Either situation is unacceptable in practice. For us the bias-variance tradeoff arises in designing the successive training units and their relationships to each other. In the end we implemented only one possibility; that is, increasing sets of training vectors were nested.

We are interested in the mse of the prediction of pixel intensity with respect to its experimental value. If we let the number $\hat{y}$ represent our prediction of the actual intensity value y (a constant for present purposes), then the mse can be expressed as a function of its variance and bias thus:

$$\begin{aligned} E[(\hat{y}-y)^2] &= E[(\hat{y}-E\hat{y}+E\hat{y}-y)^2] \\ &= E[(\hat{y}-E\hat{y})^2]+(E\hat{y}-y)^2 \\ &= \mathrm{var}(\hat{y})+(\mathrm{bias})^2 \end{aligned} \tag{2}$$

since the cross product terms vanish. When we nest and include a previous sequence in an expanded sequence, as opposed to selecting completely new random sets, we obtain a decrease in variability. For ex-

ample, if we choose a subset of training vectors that formed two images' worth of training vectors, and these vectors well represented the test sequence, the distortion could be quite low. If we then randomly choose four images' worth of training vectors from all the training vectors (not necessarily including those used for size 2), and these training vectors badly represented the test sequence, the distortion could be quite high. This potential high variability could produce an mse that would not decay as we naturally expect from an increase in the number of training vectors. However, nesting to decrease this potential variability may lead to an increase in bias that may result if one of the original smaller sequences is in some way unusual. We thus forego the opportunity to restart with a new set of training vectors and avoid such a potential bias, so that we can be faithful to the expected decay.

3 Learning Theory

Minimax results and work in progress in the theory of nonparametric function estimation suggest that, at least for numbers of training images or vectors below a threshold, the plots of distortion versus training sequence size (at a fixed bit rate) should be of the form

$$y(n) = An^{-\alpha} + B \tag{3}$$

where y = distortion and n = training sequence size. We are interested in the norm $\| y - \hat{y} \|$, where $\hat{y}$ is the prediction of pixel intensity y subject to a bit rate constraint. If we consider a particular class of distributions of pixel vector intensities, then the cited heuristic deals with the best case of the worst situation, that is, with

$$\min_{\text{predictors } \hat{y}} \quad \max_{\text{distributions}} \quad \| y - \hat{y} \|. \tag{4}$$

Previous work of Stone [6] and others deal with (what amount to) pixel intensity distributions smoother than what applies to our imaging problems. In addition, this work treats only cases for which $B = 0$, that is, where, with increasing sample size, distortion tends to 0. That is clearly impossible in our setting since, even with an infinite amount of training data, the quantizer cannot represent the original image in a distortionless way when the rate is less than the entropy of the source. And, too, we stretch to suggest that TSVQ or FVQ are algorithms that even for large samples do as well as can be in the minimax sense of (4). If the code design were indeed optimal, B could be interpreted as the operational distortion-rate function of the source for the given bit rate, code structure, and blocklength. Notwithstanding, especially in view of recent unpublished work of colleagues David Donoho and Iain Johnstone, we can make the guess that the model (3) will fit and that α will be between 0 and .5; a special argument not reproduced here suggests further that α ought to be about 1/3. The cited research of Donoho and Johnstone (unlike that of Stone) indicates that there will be a threshold above which α=1 applies. The existence of this threshold, which will be smaller at lower bit rates, renders fitting problematical when the threshold is less than the smallest sampling unit. This offers a possible explanation of the result α=3.38 for FVQ, I at 1 bpp in Table 1.

type	rate		B	A	alpha	fit
TSVQ	1 bpp	V	279	16.3	0.573	351
	1 bpp	I	248	36.0	0.245	189
	2 bpp	V	45.3	37.2	0.533	39.9
	2 bpp	I	31.1	43.7	0.366	15.7
FVQ	1 bpp	V	363	23.2	0.240	341
	1 bpp	I	380	0.028	3.38	374
	2 bpp	V	95.4	11.1	0.389	10.2
	2 bpp	I	18.4	85.7	0.035	22.6

Table 1: Fit parameters to $y(n) = B + An^{-\alpha}$

4 Results

The results for TSVQ at 1 bpp and 2 bpp are shown in Figures 1 and 2, respectively, and the results for FVQ at 1 bpp and 2 bpp are shown in Figures 3 and 4, respectively. In all figures, the x-axis has units of training sequence size (in number of images), where each image (256 x 256 pixels) provides 16384 training vectors. The y-axis has units of distortion, as measured by the mse between the input image and the quantizer output. The data points are shown as x's for the vector sampling approach, and as o's for the case of images as sampling units. The solid curve is fitted to the x's, and the dashed curve is fitted to the o's. Table 1 shows the fit parameters for the 8 cases. V and I denote the two sampling methods: vectors and images. The column marked fit shows the residual sum of squares between the fitted and actual values.

In general, the TSVQs performed better than the FVQs. This is expected because we used variable-rate TSVQ, which is designed to minimize average distortion for a given average rate rather than the more re-

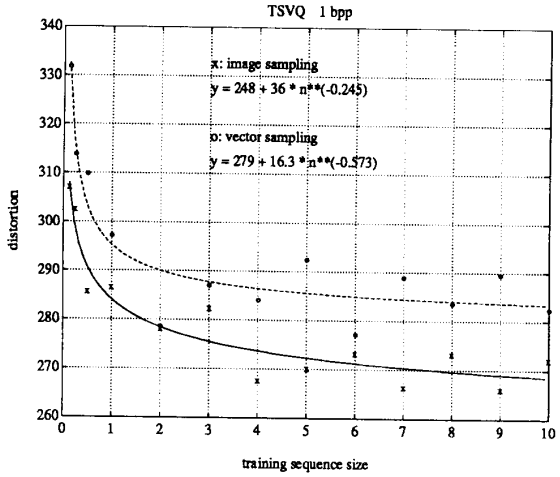

Figure 1: Distortion vs. Training Sequence Size for 1 bpp TSVQ

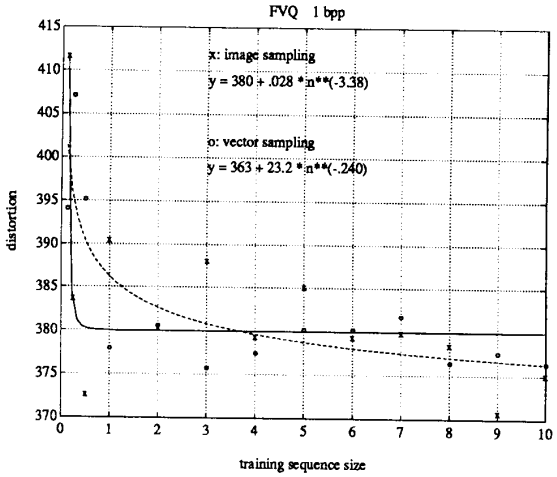

Figure 3: Distortion vs. Training Sequence Size for 1 bpp FVQ

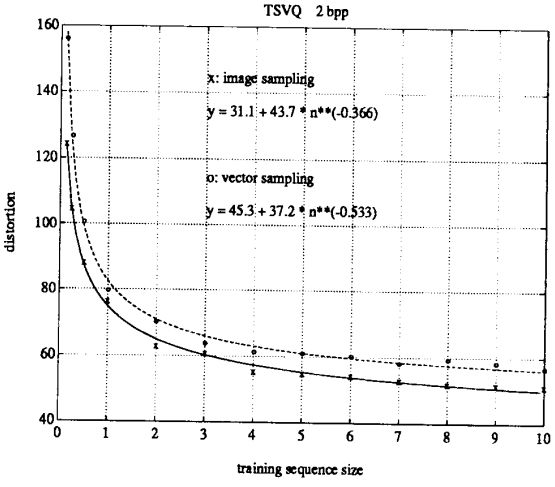

Figure 2: Distortion vs. Training Sequence Size for 2 bpp TSVQ

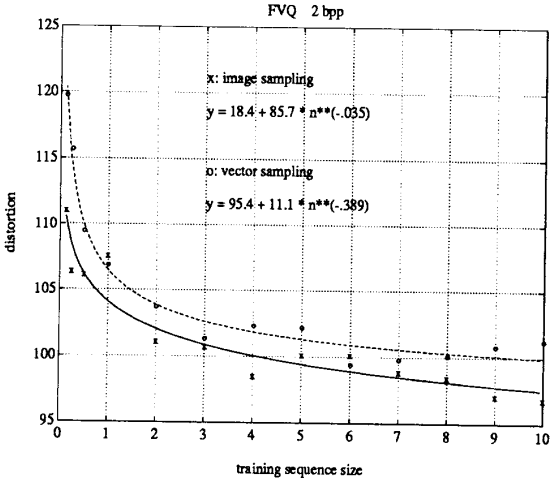

Figure 4: Distortion vs. Training Sequence Size for 2 bpp FVQ

stricted minimization of average distortion for a given fixed rate. Also as expected, the distortions at 2 bpp were considerably lower than the distortions at 1 bpp.

The differences between using images and vectors as sampling units were not substantial, although image sampling generally produced lower distortion.

The results from both TSVQ and FVQ at 1 bpp indicate that any small number of training images is sufficient for the quantizer to perform near the distortion asymptote. For 2 bpp, the results for TSVQ using both approaches and for FVQ using vector sampling fit the model of $y(n) = An^{-\alpha} + B$ very well.

For FVQ with image sampling the decay in distortion at 1 bpp was extremely rapid and, as a result, an α of about 3.4 was obtained. At 2 bpp, the distortion asymptote was much lower than expected, and α, although between 0 and 0.5, was surprisingly small. Thus, both the 1 bpp and 2 bpp results seem to indicate that the model is not applicable to FVQ with image sampling. In contrast, the A, B and α for vector sampling were reasonable values. Similar to the results for the 1 bpp TSVQ, the distortion from a training sequence size of 1 image or greater is quite close to the distortion asymptote.

The curves can be used to determine how much training data is needed to obtain results close to the asymptote for the image type considered. The experimenter first decides how much distortion above the asymptote is tolerable, and then chooses the training sequence size corresponding to that distortion.

5 Future Research and Conclusions

There are a number of issues in this work that can be further explored. Bootstrapping [2] (or some other sample reuse method) is the only plausible way to quantify the variability of estimates of A, B, α, and points on the curves. With the 10-fold cross-validation, the 10 training sequences of any particular size n were not independent. Variances computed for A, B, and α using these ten different values would thus underestimate the true variances.

The study can be extended to investigate results with different vector sizes, image sizes, and other types of images, e.g., the USC database images. In addition, the importance of variability between the images used could be considered.

The differences between the images as sampling units or vectors as sampling units can be further explored. In particular, given more training data than one is able to use, should one choose images or vectors from the set?

It is likely that the results of this preliminary study are highly dependent upon the statistics of the particular images used, yet we find it intriguing that the results fit the algebraic decay model predicted by heuristic arguments from nonparametric function estimation. Ultimately this work may provide researchers with some guidelines for both method and quantity when choosing training data for a VQ.

6 Acknowledgments

This work was supported in part by the National Institutes of Health under Grants CA49697-02 and CA55325-01, by the National Science Foundation under Grant DMS-9101548 and by Graduate Fellowships from the National Science Foundation. We thank Iain Johnstone, Jerry Halpern, and Eve Riskin for their helpful comments.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* The Wadsworth Statistics/Probability Series. Wadsworth, Belmont, California, 1984.

[2] B. Efron. *The Jackknife, the Bootstrap, and Other Resampling Plans.* Society for Industrial and Applied Mathematics, Philadelphia, 1982.

[3] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, 1992.

[4] E. A. Riskin and R. M. Gray. A greedy tree growing algorithm for the design of variable rate vector quantizers. Presented at the 1990 Picture Coding Symposium, Cambridge, MA, 1990.

[5] E. A. Riskin and R. M. Gray. A greedy tree growing algorithm for the design of variable rate vector quantizers. *IEEE Transactions on Signal Processing*, 39(11):2500–2507, 1991.

[6] C. J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10:1040–1053, 1982.