REVERSE ENGINEERING VECTOR QUANTIZERS BY TRAINING SET SYNTHESIS

Srivatsan Kandadai and Charles D. Creusere

Klipsh School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, NM 88003-88001

ABSTRACT

In this paper we present a technique for reverse engineering vector quantizers by synthesizing a training set that has similar statistics to the original training set used in designing the vector quantizer. Most VQ codebooks are designed using the LBG or generalized Lloyd algorithm which is similar to the construction of nonuniform bin histograms.Thus the VQ codebook and the number of training set vectors allocated to each of its codebook vectors is an approximation of the underlying pdf of the training set. This observation is used to synthesize a training set that has a histogram similar to the original training set. This synthesized training set can be used to construct VQs to describe subspaces of the original vector space or spaces transformed by a linear transformation.

1. INTRODUCTION

Vector quantizers are used primarily for data compression to speed transmission or reduce storage space. In many compression systems operations like linear prediction, subband decomposition or wavelet transforms are carried out prior to the use of VQ. Thus the signal that is encoded by the VQ is already represented in an efficient manner, i.e. with most of the redundancies of the source removed. The operations removing such redundancies are developed based on models of the signal source or the receivers. These models are subject to change as the understanding of the underlying systems change thereby forcing us to redesign the whole compression system. The redesign process involves training the systems again using either training set of signal or a probability density estimate that models the source of the signal. In most cases the source pdf or the original training set may not be available. Thus one has to devise a method to estimate the source model or generate a training set using the system parameters.

Most quantization can be generalized using the vector quantizer. Typically, the VQ is used as a final step in reducing the size of the signal representation dramatically. Most operations prior to the VQ are linear and deterministic, and they almost always can be inverted. These operations can be easily identified in the decoder after the inverse VQ stage. Thus if we can obtain a training set from the VQ parameters we can always reconstruct the original source training set by passing the obtained training set through the decoder after the inverse quantization stage. In this paper we describe a method to synthesize the training set of a VQ and use it to obtain VQ codebooks that operate on a transform domain of the original training set or subspaces of the original VQ training set. These subspaces or transform spaces can model the different operations that need to be modified in the new compression system. Figure 1 shows simplified versions of compression systems using the VQ. Figure 1(a) shows a compression system where the original signal is transformed and the VQ is used to compress the transformed signal[1], and Figure 1(b) shows a compression system that splits an original signal vectordirectl into different subvectors that are each compressed using a VQ.

Vector quantization (VQ) is the "ultimate" solution to the quantization of a signal vector [1]. A vector quantizer compares an input vector to a particular set of N vectors and determines which of these is "closest" (typically in a Euclidean sense). The collection of these N vectors is called the VQ codebook. The codebook entries are selected based on the nearest neighbor and centroid conditions which are necessary conditions for the codebook to be optimal for a given distribution of the inputs. The necessary conditions for optimality provide a method of iterative improvement of a given codebook. This method of iteratively designing the optimal codebook is described in the famous Linde Buzo Gray (LBG) algorithm [2].

The LBG algorithm uses an iterative clustering technique similar to the construction of nonuniform bin histograms. Thus the VQ codebook entries are equivalent to bin centers of a histogram. This idea can be used to synthesize vectors that produce a similar histogram to the original training set. This idea has been previously used for adaptive vector quantization [3][4].

2. TRAINING SET SYNTHESIS

To synthesize a good training set we require a good estimate of the shape of the pdf that generated the original trainign set. A good non-parametric estimate of the pdf obtained from a set of training data can be found by forming a histogram. A histogram measures the frequency of occurance of vectors in specific cells that partition the whole vector space, defined by the vectors of the training set. The frequency of a particular cell is an estimate of the probability mass of that cell. Thus if the cell size is reduced we end up with and estimate that converges asymptotically to the pdf of the training set [5]. Consequently, the histogram of a training set approximates the shape of the original pdf.

In our problem of synthesizing a training set, we are given only the VQ codebook vectors and, if available, the entropy codes assigned to each codebook index. To develop a methodology for synthesizing a training set for a given VQ, we look at the VQ design process. The main aim in the design of VQ is to find a codebook specifying the decoder and a partition or encoding rule, specifying the encoder, that will maximize an overall performance measure. This overall performance is specified by the statistical average over a suit-

Research sponsored by the National Science Foundation, Grant #CCR-0133115

able distortion measure. The mean square error (MSE) is used as the distortion measure here. The encoder and decoder are completely specified by the partition of the vector space $\Re^{k\times 1}$ into the cells $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N$ and the codebook, $\mathbf{C} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ respectively. Optimality for the partition satisfies the Nearest Neighbor (NN) condition, i.e. for a given set of output levels, $\mathbf{C}$, the optimal partition cells satisfy

$$\mathbf{R}_i \subset \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j); \forall j\} \tag{1}$$

and the optimal code vectors for a given partition satisfy the centroid condition given by

$$\mathbf{y}_i = cent(\mathbf{R}_i) \tag{2}$$

For the squared error distortion measure, the centroid of a set $\mathbf{R}_i$ is simply the minimum mean squared estimate of $\mathbf{X}$ given $\mathbf{X} \in \mathbf{R}_i$, i.e.

$$\mathbf{y}_i = cent(\mathbf{R}_i) = E(\mathbf{X}|\mathbf{X} \in \mathbf{R}_i) \tag{3}$$

The above mentioned conditions are the necessary (but not sufficient) conditions for optimality of a VQ codebook and partition. Thus by iteratively using the two conditions one can obtain a VQ design that is at least locally optimal.

This VQ design procedure is similar to constructing a variable bin histogram [2]. A variable bin histogram is generated by assuming m cells $A_1, A_2, \ldots, A_m$ charecterized by the coordinate of the center x_i and the number of samples within each cell k_i [6]. In the VQ case, cells are charecterzed by the VQ codebook entries and the partition rules. The only thing missing is the number of samples within each cell. This however can be calculated from the entropy codes assigned to each codebook entry, since the entropy codes depend on the probability of the code vectors. If the VQ designed is truly optimal, we can assume that each of the code vectors is equally probable and assign an equal number of training set elements to each codebook partition. Thus the probability density in each cell is uniform. The shape of the underlying pdf of the training set is thus preserved in the spacing of the cells and their shapes. The larger a cell, the lower the probability density; the smaller the cell the higher the density.

Thus the training set can be synthesized by generating uniformly distributed vectors for each VQ partition with the centroid as the mean and the distance of the nearest cell boundary as the maximum deviation from the mean. The number of vectors per cell can vary as dictated by the probabilities of the code vectors estimated from the entropy codes, or can be the same for each cell if the code vectors are equally probable. When we do not have the entropy code information, we can assume that each cell has equal number of vectors. This synthesized training set will have a histogram similar to the original training set and the VQ will be locally optimal for the synthesised training set. In Figure 2, the first plot shows a scatter plot of a training set (denoted by 'o') and the VQ designed (denoted by 'x'), the second plot shows the synthesized training set constructed from the VQ codebook vectors when the probabilities of each code vector indices are known and the third plot shows the synthesized training set assuming that each of the code vectors are equally probable. Comparing these plots, we see that the synthesized training set data in Figures 2(b) and 2(c) are scattered similarly to the original training set data as shown in Figure 2(a). The following enumerates the different steps used in the synthesis of a training set from a given VQ codebook.

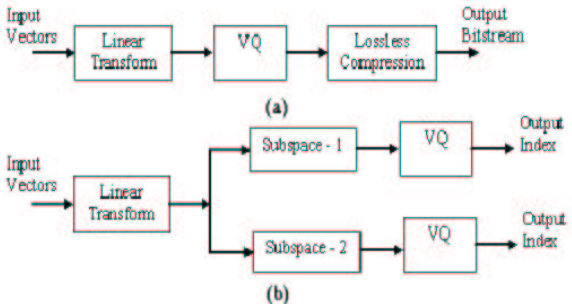

Figure 1: Block diagrams of general VQ based compression systems. (a) a system that quantizes the transformed signal using a VQ, (b) a system that quantizes subspaces of the original signal separately.

- Assume that the original training set is large compared to the VQ code book, and that the number of elements in it is N. If the entropy codes are known we can estimate the probability of a codebook vector, $p(\mathbf{y}_i)$ since it is inversely proportional to the number of bits assigned to the code vector. This probability estimate multiplied by N gives the number of vectors in a partition. If the the entropy codes are not given we assume that the number of vectors in each partition cell is equal, given by N/k where k is the number of code vectors in the codebook.
- For a given codebook vector $\mathbf{y}_i$, after estimating the number of elements say n_i in its partition region, a pseudo random generator is used to generate n_i uniformly distributed samples with mean $\mathbf{y}_i$ and the distance to nearest partition boundary as the maximum deviation.
- The previous step is repeated for all the codebook vectors.

In the above discussion we generate training sets assuming that the Euclidean distance measure is used. Euclidean distance or the L_2 norm is a metric defined as

$$d = (\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2) \tag{4}$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \Re^{n\times 1}$, and d is the squared distance between $\mathbf{x}_1$ and $\mathbf{x}_2$. In many cases weighted distance measures are used to make to make the VQ a more efficient signal space. A weighted distance measure is given by

$$d_w = (\mathbf{x}_1 - \mathbf{x}_2)^T\mathbf{D}(\mathbf{x}_1 - \mathbf{x}_2) \tag{5}$$

where $\mathbf{D} \in \Re^{n\times n}$. A major property of a distance metric is that it is always nonegative, i.e.

$$d_w = (\mathbf{x}_1 - \mathbf{x}_2)^T\mathbf{D}(\mathbf{x}_1 - \mathbf{x}_2) \geq 0 \tag{6}$$

which implies that $\mathbf{D}$ is always positive semidefenite and so $\mathbf{D}$ can be factored into,

$$\mathbf{D} = \mathbf{R}^T\mathbf{R} = \mathbf{U}^T\Lambda^{1/2}\Lambda^{1/2}\mathbf{U} \tag{7}$$

where $\mathbf{U}$ is a unitary matrix of eigenvectors of $\mathbf{D}$ as its columns and Λ is a diagonal matrix with the eigenvalues of the $\mathbf{D}$ as the diagonal entries. Thus the weighted distance measure can be written as

$$d_w = (\mathbf{x}_1 - \mathbf{x}_2)^T\mathbf{R}^T\mathbf{R}(\mathbf{x}_1 - \mathbf{x}_2) \tag{8}$$

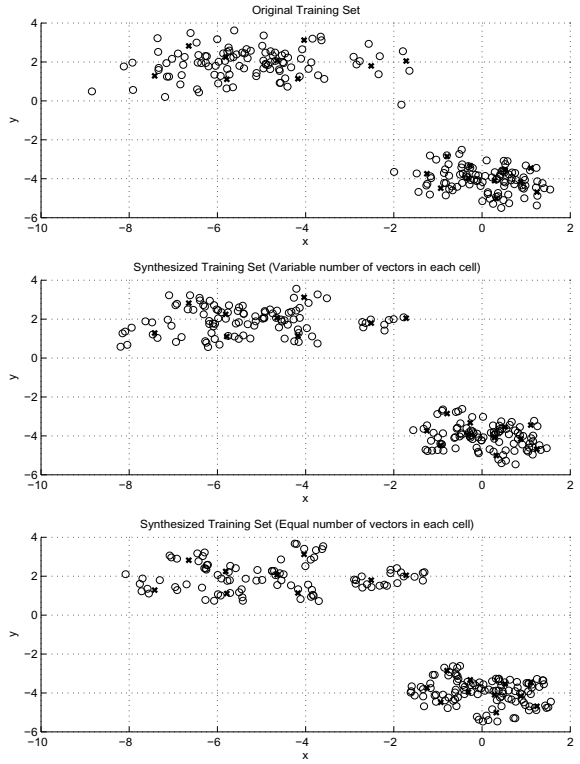

Figure 2: Scatter plots of the original and synthesized training sets

which is nothing but the Euclidean distance in the space transformed by the matrix $\mathbf{R}$. Therefore, to synthesize the training set for a VQ designed using a weighted distance measure, all we have to do is to construct the training set as discussed above and transform all the synthesized vectors by the matrix $\mathbf{R}^{-1}$. We know $\mathbf{R}^{-1}$ exists because $det(\mathbf{R}) \neq 0$.

3. EXPERIMENTAL RESULTS

The main goal of this work is to design VQs that define subspaces and tranformed spaces of the original training set space. We present three experiments, one to recreate VQs for the subspaces of the original training set, the second one recreates a VQ on the DCT of the original training set and the third experiment is for VQs designed using a weighted distortion metric. The measure used to compare the performance of the VQs is the squared error averaged over the values from the original training sets. The initial code books used for VQ design using the original and the synthesized training sets are the same.

3.1 Synthesizing subspace quantizers

In this experiment, we design VQs using the synthesized training set to discribe different subspaces that can be formed by grouping together different elements in the original vector space. If $\mathbf{x}$ is a vectors such that $\mathbf{x} \in \Re^n \times 1$, with elements $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, subvectors can be formed by grouping subsets of these elements into smaller vectors. Like $\mathbf{x}_1 = [x_1, \ldots, x_l]$ and $\mathbf{x}_1 = [x_{l+1}, \ldots, x_n]$. Two experiments are presented here. In the first experiment, a five-dimensional Gaussian with zero mean and varance of $\sigma = \mathbf{I}$ (the identity matrix) is used to generate the original training set of 1000 vectors. From this training set we design a VQ with a codebook containing 20 codebook vectors. Using this codebook, we then synthesize a new training set, and use it to design VQs for the subspace defined by the first two and the last three elements of the original five dimensional vector space. The error performance of the VQs are then compared to the performance of the VQs designed for the subspace vectors from the original training set. The performance measure used is the mean square error (MSE), and the results are as shown in Table 1. The initial codebook for the design of all the VQs is the same for a given vector space. Table 2 shows the MSE performance of VQs for a different initial training set. Here the the vector space is four dimensional and the original training set is generated by a bimodal gaussian distribution with means $[\ 0 \quad 0 \quad 0 \quad 0\ ]^T$ and $[\ 2 \quad 3 \quad 2 \quad 1\ ]^T$, and variances $\sigma_1 = \mathbf{I}$ (identity) and

$$\sigma_2 = \begin{bmatrix} 1 & 0.3 & 0.1 & 0 \\ 0.3 & 1 & 0.3 & 0.1 \\ 0.1 & 0.3 & 1 & 0.3 \\ 0 & 0.1 & 0.3 & 1 \end{bmatrix}$$

The subspace vectors are formed by combining the first two and the last two elements of the original four dimensional vector.The results show that the VQs designed using the synthesized training set produce almost the same results as the VQs designed from the original training set.

Table 1: Comparison of performance between VQs designed using the original training set and the sythesized training sets.

Subspace	Original performance (MSE)	Synthesized performance (with entropy) (MSE)	Synthesized performance (without entropy) (MSE)
1	0.23	0.25	0.26
2	0.32	0.36	0.38

Table 2: Comparison of performance between VQs designed using the original training set and the sythesized training sets for a bimodal gaussian distribution.

Subspace	Original performance (MSE)	Synthesized performance (with entropy) (MSE)	Synthesized performance (without entropy) (MSE)
1	0.31	0.33	0.34
2	0.29	0.31	0.32

3.2 Synthesizing transformed space quantizers

In this experiment we demonstrate how to design vector quantizers that operate on a space transformed by a linear transformation. The original training set of one thousand samples, is generated by a four dimensional gaussian distribution with mean $[\ 0 \quad 0 \quad 0 \quad 0\ ]^T$ and variance $\sigma = \mathbf{I}$. The VQ for this training set (VQ-original), and it has 20 codebook vectors. The VQ of the DCT of the training set, VQ-DCT is also calculated. Our goal here is to design a VQ

Table 3: Comparison of performance between VQs designed using the original training set and the sythesized training set for linearly transformed data.

Space	Original performance (MSE)	Synthesized performance (with entropy) (MSE)	Synthesized performance (without entropy) (MSE)
DCT	0.77	0.79	0.79
A	0.76	0.0.79	0.82

using the VQ-original codebook, that performs close to VQ-DCT in the DCT domain. To do so we first synthesize the training set from VQ-original and perform the DCT on the synthesized training set to obtain a DCT domain training set. This is then used to calculate a VQ that perfoms on the DCT domain. This experiment is repeated for an arbitrary matrix transform given by

$$\mathbf{A} = \begin{bmatrix} 8 & -5 & 2 & 11 \\ 6 & 5 & 7 & 9 \\ 1 & 3 & 10 & 4 \\ 4 & 1 & 9 & 7 \end{bmatrix}$$

Table 3 compares the MSE performance of the VQs designed on the original training set with that designed on the synthesized training set. The MSEs obtained are very close.

The training set synthesis method does not always work. If the given VQ is a very sparse representation of the original training set then the synthesized VQs do not perform very well. For the same training set above if we design a VQ with only five codebook entries then the differences in the original and synthesized VQ performance is large. Certain VQ codebook vectors end up representing a very sparse set of points of the original training set, which when used to synthesize the codebook generates a large number of samples in that region. The best way to ensure such errors are reduced is to use the entropy codes to estimate the probability of each codeword and then generate samples for each partition accordingly.

3.3 Synthesizing quantizers from VQs that use weighted distance measures

This is similar to the method of finding VQs of systems transformed by linear transforms as discussed previously. Here we generate a four dimensional training set with a gaussian distribution having zero mean and identity variance matrix. A VQ is designed using a weighted distance measure as discribed in equation (6) with

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.81 & 0 & 0 \\ 0 & 0 & 0.49 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$

This VQ is then used to synthesize the training set assuming that the distances are Euclidiean as in (4), and is transformed by the matrix

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Table 4: Comparison of performance between VQs designed using the original training set and the sythesized training set for VQs designed using a weighted distortion measure.

subpspace	Original performance (MSE)	Synthesized performance (with entropy) (MSE)	Synthesized performance (without entropy) (MSE)
R^{-1}	0.33	0.34	0.37

which is obtained according to equation (7). Table 4 compares the VQs generated by the synthesized and the original training sets transformed by the matrix $\mathbf{R}$.

4. CONCLUSION

In this paper, we have presented a method for reverse engineering VQs to obtain a training set that has statistical properties similar to those of the original training set. This synthesized training set can be used to do many things like designing VQs to describe subspaces of the original VQ space and designing VQs that can be used in a linear transform domain space. We can also use this technique to obtain original source training sets for systems whose end stage consists of a VQ, by simply passing the synthesized training set through the stages of the decoder that come after the VQ-decoder stage.

This technique may work better if some kind of an estimate of the probabilities of the code vector can be made, say by using the entropy codes which have been applied in the codec to further compress the VQ indices – i.e., the probability of a code vector is a measure of the probability mass of that vector's quantization cell and it gives a better estimate of the shape of the pdf of the original training set.

REFERENCES

[1] Allen Gersho and Robert M. Gray, *Vector Quantizaion and Signal Compression*. Kluwer Academic Publishers, Massachusetts 02061 USA.

[2] Y.Linde, A.Buzo, and R.M.Gray, "An algorithm for vector quantizer design," in *IEEE Trans. on Commun.*, Vol. COM-25, 1980, 84-95.

[3] D. Comaniciu, "Training Set Synthesis For Entropy-Constrained Transform Vector Quantization," in *Proc. ICASSP 1996*, Vol. 4, 7-10 May 1996, pp. 2036-2039.

[4] D. Comaniciu, "Model-Based Training Set Synthesis for Vector Quantization," in *Proc. of IASTED International conference on Signal and Image Processing*, Nassau, Bahamas, Oct. 18-12, 1999.

[5] R.O. Duda, P.E.Hart and D.G. Stork, *Pattern Classification*. Second Edition, John Wiley & Sons, Inc., New York, USA.

[6] Keinosuke Fukunaga, *Introduction To Statistical Pattern Recognition*. Academic Press Inc. NY 10003.