A GENERALIZED VQ METHOD FOR COMBINED COMPRESSION AND ESTIMATION

Ajit Rao, David Miller, Kenneth Rose, and Allen Gersho

Center for Information Processing Research
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106

ABSTRACT

In vector quantization, one approximates an input random vector, $\mathbf{Y}$, by choosing from a finite set of values known as the codebook. We consider a more general problem where one may not have direct access to $\mathbf{Y}$ but only to some statistically related random vector $\mathbf{X}$. We observe $\mathbf{X}$ and would like to generate an approximation to $\mathbf{Y}$ from a codebook of candidate vectors. This operation, called *generalized vector quantization* (GVQ), is essentially that of *quantized estimation.* An important special case of GVQ is the problem of noisy source coding wherein a quantized approximation of a vector, $\mathbf{Y}$, is obtained from observation of its noise-corrupted version, $\mathbf{X}$. The optimal GVQ encoder has high complexity. We overcome the complexity barrier by optimizing a structurally-constrained encoder. This challenging optimization task is solved via a probabilistic approach, based on deterministic annealing (DA), which overcomes problems of shallow local minima that trap simpler descent methods. We demonstrate the successful application of our method to the coding of noisy sources.

1. GENERALIZED VECTOR QUANTIZATION

Consider the problem of estimating a random vector, $\mathbf{Y}$ from a statistically related vector, $\mathbf{X}$. If the estimate $V(\mathbf{X})$ is constrained to take on values fro a finite set of N "estimation vectors", the mapping from $\mathbf{X}$ to $\mathbf{Y}$ is called *generalized vector quantization* (GVQ). Note that GVQ reduces to ordinary VQ in the special case where $\mathbf{X} = \mathbf{Y}$.

A wide range of applications in source coding can be formulated as GVQ problems, including: (a) noisy source coding problems where $\mathbf{Y}$ is the signal of interest, but only a noise-corrupted version, $\mathbf{X}$, is available to the encoder; (b) complexity-constrained source coding problems wherein $\mathbf{Y}$ has a large dimensionality and must be replaced by lower dimensional "feature" vectors $\mathbf{X}$ that are extracted from $\mathbf{Y}$ prior to quantization.

This work was supported in part by the National Science Foundation under grant no. NCR-9314335, the University of California MICRO program, DSP Group, Inc. Echo Speech Corporation, Moseley Associates, National Semiconductor Corp., Qualcomm, Inc., Rockwell International Corporation, Speech Technology Labs, and Texas Instruments, Inc.

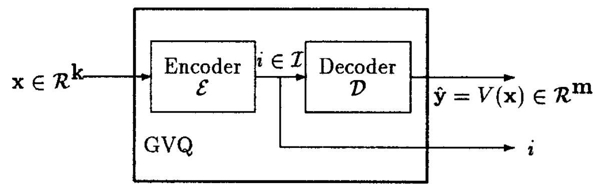

Figure 1: Block diagram of a Generalized Vector Quantizer

While a quantized estimator is mandated for data compression problems by the rate constraint, for other estimation problems, quantization may also be usefully employed to limit complexity and to facilitate implementability of the estimator, as the optimal estimator may be difficult to evaluate and may require a large complexity of specification. For N sufficiently large, a quantized estimator will closely approximate the optimal estimator, while achieving a practical implementation.

Formulation

We treat the mapping of a feature vector, $\mathbf{X}$, to a finite-valued estimate of a vector $\mathbf{Y}$ as a single operation, called *generalized vector quantization* (GVQ). Mathematically, a *generalized vector quantizer* V with input dimension k, output dimension m, and size N is a mapping $V : \mathcal{R}^k \to \mathcal{C}$, where $\mathcal{C} \equiv \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\} \subset \mathcal{R}^m$ is the *codebook* with size $|\mathcal{C}| = N$. The elements of $\mathcal{C}$ are the *estimation vectors.* We also define the corresponding set of *codevector indexes* $\mathcal{I} \equiv \{1, 2, \cdots, N\}$. Effectively, the GVQ defines a *partition* of the input space, $\mathcal{R}^k$ into N regions, R_i for $i \in \mathcal{I}$. The partition regions are defined as:

$$R_i = \{\mathbf{x} \in \mathcal{R}^k : V(\mathbf{x}) = \mathbf{y}_i\}. \tag{1}$$

The GVQ performance is measured in terms of the expected distortion

$$D = E[d(\mathbf{Y}, V(\mathbf{X}))], \tag{2}$$

where the distortion measure $d(\mathbf{y}, \hat{\mathbf{y}})$ is the cost of approximating $\mathbf{y}$ by $\hat{\mathbf{y}}$. In principle, a GVQ is fully characterized by specifying (a) the feature space partition and (b) the codebook $\mathcal{C}$. Correspondingly, the GVQ system, as shown in Figure 1, can be viewed as the composition of two operations, an *encoder*, $\mathcal{E}$, which assigns an index $i \in \mathcal{I}$ to each input vector $\mathbf{x}$, and a *decoder*, $\mathcal{D}$, which is a table-lookup operation that produces the estimation vector, $\mathbf{y}_i$ corresponding to the index generated by the encoder. However, unlike ordinary VQ where $\mathbf{X} = \mathbf{Y}$, in the general case of GVQ, $\mathbf{X}$ and $\mathbf{Y}$ may have different dimensionalities and only a partially known statistical relationship. Consequently, the encoder does not try to minimize a distortion with respect to $\mathbf{X}$, but instead functions as a classifier whose performance measure is the distortion in $\mathbf{Y}$ induced by the classification.

The optimal GVQ must satisfy necessary conditions similar to those satisfied by optimal VQ.

(a) Encoder:
$R_i \equiv \{\mathbf{x} : E[d(\mathbf{Y}, \mathbf{y}_i) \,|\, \mathbf{X} = \mathbf{x}] \leq E[d(\mathbf{Y}, \mathbf{y}_j) \,|\, \mathbf{X} = \mathbf{x}], \forall j \in \mathcal{I}\}$,

(b) Decoder:
$\mathbf{y}_i = \arg \min_{\mathbf{y} \in \mathcal{R}^m} \{E[d(\mathbf{Y}, \mathbf{y}) \,|\, \mathbf{X} \in R_i]\}$.

In the case of the squared error distortion measure, these conditions specialize to :

(a) Encoder:
$R_i \subset \{\mathbf{x} : \|G(\mathbf{x}) - \mathbf{y}_i\|^2 \leq \|G(\mathbf{x}) - \mathbf{y}_j\|^2, \forall j \in \mathcal{I}\}$.

(b) Decoder:
$\mathbf{y}_i = E[G(\mathbf{X}) | \mathbf{X} \in R_i]$.

where $G(\mathbf{x}) \equiv E[\mathbf{Y}|\mathbf{X} = \mathbf{x}]$ is the conditional estimator of $\mathbf{Y}$ given $\mathbf{X}$. For the squared-error distance, optimal GVQ can be implemented as the cascade of the conditional estimator, $G(\mathbf{x})$ with the optimal vector quantization of this estimate. While this result is elegant conceptually and has been suggested by several authors including [2], [3], [4], [5], [6], it assumes knowledge of the optimal estimator, G, which may be either unavailable or too complex to represent. Moreover, the optimality of estimation followed by quantization was only obtained for the special case of squared error distortion. In the more general case, one must revert to the encoder-decoder structure of Figure 1, where significant implementation difficulty comes from the complex nature of the encoder partition cells. Unlike the case of VQ, the GVQ encoding rule does not enforce a Voronoi partition of the feature space, but rather allows general, highly complex decision regions which imply concomitant high encoder search complexity. Evidently, we must limit the complexity of the encoder to obtain an implementable solution. We do this by imposing structural constraints on the partition regions. In practice, the encoder can be implemented by any of the well-known pattern classifier structures. We propose to use a multiple-prototype classifier structure which we next describe.

2. MULTIPLE-PROTOTYPE CLASSIFIER

In a *multiple-prototype classifier*, each of the N encoder regions, R_j "owns" a set of prototypes, $\mathbf{x}_{jk} \in \mathcal{R}^k$ for $k = 1, 2, ..., M_j$. The input, $\mathbf{x}$ is compared to all prototypes using a distance measure,$d_e(\cdot)$, and the class index of the nearest prototype is selected :

$$i = \arg \min_j \{\min_k [d_e(\mathbf{x}, \mathbf{x}_{jk})]\}, \qquad (3)$$

The prototypes define Voronoi (nearest-neighbor) cells $\mathcal{C}_{jk}$ in the input space and provide the support for the encoder partition. The encoder partition region R_j is the union of M_j Voronoi cells:

$$R_j = \bigcup_{k=1}^{M_j} \mathcal{C}_{jk}. \qquad (4)$$

One may interpret this approach as trying to approximate the optimal non-convex partition region by a union of Voronoi cells. We refer to a GVQ system based on such an encoder as multiple-prototype generalized vector quantizer (MP-GVQ).

By imposing the MP-GVQ structure, we have simplified the *implementation* problem, with some loss of optimality. However, the *design* is a hard optimization problem. In the next section, we look into MP-GVQ design and propose the use of deterministic annealing to overcome the challenging nature of the optimization problem.

3. MP-GVQ DESIGN

The MP-GVQ design problem is that of *jointly* optimizing the prototypes $\{\mathbf{x}_{jk}\}$ and code vectors $\{\mathbf{y}_j\}$ to minimize the MP-GVQ distortion. More precisely, given a training set $\mathcal{T} = \{(\mathbf{x}, \mathbf{y})\}$, we must find the optimal sets of prototypes $\{\mathbf{x}_{jk}\}$ and code vectors $\{\mathbf{y}_j\}$ to minimize the overall distortion :

$$\min_{\{\mathbf{x}_{jk}\}, \{\mathbf{y}_j\}} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} d(\mathbf{y}, V(\mathbf{x})), \qquad (5)$$

Here, V is the GVQ function, $V(\mathbf{x}) = \mathbf{y}_i$, where i is the encoder index satisfying (3).

It is easy to implement optimality condition (b) and directly optimize the decoder for a fixed encoder. The main difficulty lies in optimizing the encoder given a fixed decoder. There is no direct way to apply the encoder optimality condition while satisfying the MP-GVQ structural constraint. We first describe a simple suboptimal design procedure and then motivate and introduce the deterministic annealing method for design.

Extended Nonlinear Interpolative VQ (ENLIVQ)

Nonlinear interpolative vector quantization (NLIVQ) was introduced as a quantized estimation technique in [7] and has since found several applications in speech and image coding. In the Extended NLIVQ approach, we generalize these ideas by allowing multiple prototypes per encoder partition region. The locations of the prototypes are initially chosen using a judicious heuristic. This defines Voronoi cells that form the basis for the encoder partition. Next, an iterative design of the decoder and the mapping function between the Voronoi cells and the outputs is carried out. This second step is guaranteed to descend in the GVQ cost and forms the encoder partition regions as unions of the Voronoi cells. However, one cannot re-optimize the locations of the prototypes (and hence the Voronoi cells) once they have been initially chosen. Improved performance can only be obtained by allowing more prototypes, which increases the complexity of the GVQ and also makes it less robust. Thus, there is strong motivation to directly attack the ultimate problem of jointly optimizing the encoder partition and the decoder. In the next section, we develop a method which not only performs a joint optimization but also attempts to avoid non-global optima via a deterministic approximation to an annealing process.

Deterministic Annealing

We adopt an approach inspired by ideas from information theory and statistical physics. This work is based on the deterministic annealing (DA) approach to VQ design [8] and extends it for GVQ design. The main extension is in imposing structural constraints as will be explained below. This leads to a powerful optimization method which is robust to local minima. In a related work in pattern recognition, we have recently applied the structurally constrained DA approach to the problem of statistical classifier design and obtained consistent, substantial improvements over standard methods [9].

We cast the problem within a probabilistic framework, where we consider a "random" GVQ encoder, characterized by a probabilistic assignment of the input data to the Voronoi cells of the encoder. Accordingly, we define the *probabilities of "association"* $\{P[\mathbf{x} \in C_{jk}]\}$, where we emphasize that $\mathbf{x}$ is the observable data available to the encoder. The probabilistic associations of data with the encoder regions are then $\{P[\mathbf{x} \in R_j]\} = \sum_k P[\mathbf{x} \in C_{jk}]$. Further, we impose the structural constraint on the encoder partition by appropriately choosing a parametrization of the association probabilities. (A similar "trick" was used earlier in the context of tree-structured clustering [10].) For the problem at hand, an appropriate choice is the Gibbs distribution

$$P[\mathbf{x} \in C_{jk}] = \frac{e^{-\gamma d_e(\mathbf{X},\mathbf{X}_{jk})}}{\sum_{l,m} e^{-\gamma d_e(\mathbf{X},\mathbf{X}_{lm})}}, \tag{6}$$

which is parameterized to enforce the MP-GVQ structure. This choice can also be shown to follow directly from the maximum entropy principle. Here, γ is a positive "scale" parameter which controls the fuzziness of the distribution. For finite values of γ, the distribution represents a probabilistic association of an input $\mathbf{x}$ with the cells C_{jk}, based on the distance to the corresponding prototypes. Closer prototypes have higher likelihood of association. For $\gamma \to \infty$, $\{P[\cdot] \to \{0, 1\}\}$, *i.e.*, the "random" GVQ becomes a deterministic quantizer, using the encoding rule of (3).

The expected distortion of a "random" GVQ based on the Gibbs encoder above is :

$$D = \sum_{(\mathbf{X},\mathbf{Y})\in \mathcal{T}} \sum_j P[\mathbf{x} \in R_j] d(\mathbf{y}, \mathbf{y}_j). \tag{7}$$

The entropy of this distribution,

$$H = -\sum_{(\mathbf{X},\mathbf{Y})\in \mathcal{T}} \sum_j \sum_k P[\mathbf{x} \in C_{jk}] \log P[\mathbf{x} \in C_{jk}], \tag{8}$$

characterizes the degree of randomness of the solution. The information-theoretic approach suggests that while the final GVQ solution that we expect is non-random, it is advantageous for avoiding shallow local optima, to impose this determinism gradually. The idea is to control the randomness by gradually reducing entropy while minimizing the expected GVQ distortion. Note that the minimization is performed with respect to the decoder reproductions $\{\mathbf{y}_j\}$, as well as the distribution parameters γ and $\{\mathbf{x}_{jk}\}$.

Solving the constrained optimization problem is equivalent to solving the unconstrained minimization of the Lagrangian, *i.e.*

$$\min_{\{\mathbf{x}_{jk}\},\gamma,\{\mathbf{y}_j\}} L \equiv \min_{\{\mathbf{x}_{jk}\},\gamma,\{\mathbf{y}_j\}} D - \frac{1}{\beta} H, \tag{9}$$

where β is the Lagrange multiplier used to enforce the entropy constraint. For $\beta = 0$, the sole objective is entropy maximization, which is achieved by the uniform distribution, i.e. by choosing the prototype vectors to be non-distinct. For $\beta \to \infty$, minimizing L is equivalent to minimizing the distortion, leading to a non-random (i.e. $H \to 0$) generalized vector quantizer. This solution is obtained within our probabilistic framework by choosing all prototype vectors to be distinct and sending $\gamma \to \infty$. Further, the Lagrangian described in (9) is analogous to the Helmholtz free energy of a physical system in thermal equilibrium. For such a system, D represents the energy and H the macroscopic entropy of the system in thermal equilibrium at temperature $\frac{1}{\beta}$. This analogy and the other observations made about the cost function motivate us to use an annealing approach to minimize the Lagrangian, L, starting from $\beta = 0$ and tracking the solution while increasing β towards infinity. In this way, we obtain a sequence of solutions of decreasing entropy and distortion, leading to a "hard" MP-GVQ in the limit. The annealing process can help to avoid local optima of the cost.

4. GVQ DESIGN FOR NOISY SOURCES

The objective of the noisy source coding problem is to design a quantizer for a random vector $\mathbf{Y}$, given a noise-corrupted form of the vector, $\mathbf{X}$. Ephraim and Gray have proposed a "modified distortion measure" approach, using the fact that (for certain distortion measures) the optimal noisy source quantizer can be implemented by cascading the optimal estimator, $G(\mathbf{X})$, with the optimal quantizer for the estimator [2]. However, their approach assumes that the optimal estimator is a simple, known function. In our approach, we do not assume knowledge of the optimal estimator since the function may be unknown or may be too complex for direct modeling. Instead, we design a noisy source coder given only the knowledge embodied in a training set of $(\mathbf{x_t}, \mathbf{y_t})$ pairs. One could use the entire training set as a look-up table to approximate the optimal estimator. However, such an estimate would have unmanageable complexity, and would likely not generalize well outside the training set. This motivated us to use a lower complexity MP-GVQ solution designed via the DA method. As will be demonstrated, our approach has the added advantage of robustness.

We experimented with two different noisy sources. It is important to note that, although we specify how the sources were generated, this information was not made available to the GVQ design method. In the first example we chose a first-order Gauss-Markov source as the clean signal, and corrupted it by additive Laplacian noise with the same variance. In the second example, we chose the same Gauss-Markov source as the clean signal, but corrupted it with a multiplicative noise process. In this case, the noisy source is given by the equation

$$y(n) = [1 + \epsilon w(n)]x(n) + v(n), \tag{10}$$

where both $w(n)$ and $v(n)$ are Gaussian noise processes with zero mean and unity variance. In both examples, $y(n)$ and $x(n)$ were grouped into non-overlapping four-dimensional vectors to form the training set. For each example, we used the DA method to generate two GVQ solutions, one with a decoder codebook of size N=16 and the other with size N=64. The training set (TRS) consisted of 5000 vectors. The performance of the GVQ was also tested on a test set (TSS) of 50000 vectors not used during the design process. A low-complexity, single prototype per decoder reproduction was used for all the DA solutions. The solutions were compared with MP-GVQs designed using the ENLIVQ method described earlier, wherein a larger number of prototypes were allowed per reproduction.

Our results in Table 1 and Table 2 show that the ENLIVQ method required a significant increase in encoder complexity to achieve the same training set performance as the MP-GVQs designed via the DA approach. Moreover, on the test set, the ENLIVQ solutions, using a significantly larger number of prototypes, did not perform as well as DA solutions that used substantially fewer prototypes. In fact, the performance dropped with increasing complexity, as one would expect. In the limiting case where the entire training set was used as a look-up table encoder, there was a dramatic difference between training and test set performance, and the test set performance was particularly poor.

5. CONCLUSIONS

In conclusion, the GVQ method that we have suggested achieved considerable improvements over existing methods for the compression of noisy sources. The method holds particular promise for the compression of noisy images.

6. REFERENCES

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

[2] Y. Ephraim and R. M. Gray. A unified approach for encoding clean and noisy sources by means of waveform and autoregressive model vector quantization. *IEEE Transactions on Information Theory,* 34:826–834, 1988.

[3] D. Sakrison. Source encoding in the presence of random disturbance. *IEEE Trans. on Inform. Theory,* IT-14:165–166, 1968.

[4] T. Fine. Optimum mean-square quantization of a noisy input. *IEEE Trans. on Inform. Theory,* IT-11:293–294, 1965.

[5] R. L. Dobrushin and B. S. Tsybakov. Information transmission with additional noise. *IEEE Trans. on Inform. Theory,* IT-8:293–304, 1962.

[6] J. K. Wolf and J. Ziv. Transmission of noisy information to a noisy receiver with minimum distortion. *IEEE Trans. on Inform. Theory,* IT-16:406–411, 1970.

[7] A. Gersho. Optimal nonlinear interpolative vector quantization. *IEEE Trans. on Comm.,* COM-38:1285–1287, 1990.

[8] K. Rose, E. Gurewitz and G. C. Fox. Vector quantization by deterministic annealing. *IEEE Trans. on Inform. Theory,* IT-38:1249–1258, 1992.

[9] D. Miller, A. Rao, K. Rose, and A. Gersho. A Global Optimization Technique for Statistical Classifier Design. submitted to *IEEE Transactions on Signal Processing.*

[10] D. Miller and K. Rose. A non-greedy approach to tree-structured clustering. *Patt. Rec. Lett.,* 15:683–690, 1994.

Method	DA	ENLIVQ				
M	16	16	32	64	96	112
SNR(TRS)	4.99	4.40	4.54	4.79	4.99	5.03
SNR(TSS)	4.44	4.26	4.39	4.55	4.58	4.63

(a) N=16

Method	DA	ENLIVQ					
M	64	64	128	192	256	320	384
SNR(TRS)	6.05	4.99	5.34	5.48	5.69	5.79	6.08
SNR(TSS)	4.98	4.70	4.78	4.77	4.77	4.74	4.68

(b) N=64

Table 1: SNR comparisons of the ENLIVQ and DA methods for a Gauss-Markov source corrupted by Laplacian noise. N is the size of the codebook, M is the number of prototypes used, TRS represents the training set and TSS represents the test set.

Method	DA	ENLIVQ					
M	16	16	32	64	128	256	5000
SNR(TRS)	8.07	7.38	7.18	7.32	7.44	7.72	10.17
SNR(TSS)	7.78	7.45	7.22	7.28	7.36	7.53	6.39

(a) N=16

Method	DA	ENLIVQ					
M	64	64	128	256	512	768	960
SNR(TRS)	8.82	8.37	8.39	8.59	8.82	9.06	9.2
SNR(TSS)	8.21	8.15	8.17	8.19	8.19	8.13	8.06

(b) N=64

Table 2: SNR comparisons of the ENLIVQ and DA methods for a Gauss-Markov source corrupted by multiplicative noise. N is the size of the codebook, M is the number of prototypes used, TRS represents the training set and TSS represents the test set.