VECTOR-QUANTIZATION BY DENSITY MATCHING IN THE MINIMUM KULLBACK-LEIBLER DIVERGENCE SENSE

Anant Hegde[1], Deniz Erdogmus[1], Tue Lehn-Schioler[2], Yadunandana N. Rao[1], Jose C. Principe[1]
[1] CNEL, Electrical & Computer Engineering Department, University of Florida, Gainesville, Florida, USA
[2] Intelligent Signal Processing, Informatics and Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark

Abstract – Representation of a large set of high-dimensional data is a fundamental problem in many applications such as communications and biomedical systems. The problem has been tackled by encoding the data with a compact set of code-vectors called processing elements. In this study, we propose a vector quantization technique that encodes the information in the data using concepts derived from information theoretic learning. The algorithm minimizes a cost function based on the Kullback-Liebler divergence to match the distribution of the processing elements with the distribution of the data. The performance of this algorithm is demonstrated on synthetic data as well as on an edge-image of a face. Comparisons are provided with some of the existing algorithms such as LBG and SOM.

I. INTRODUCTION

Encoding an information source with a smaller set of code vectors is a fundamental problem in digital signal processing. There exists a huge literature on vector quantization (VQ) algorithms that use various cost functions to minimize the average distortion between the dataset and the information contained in the codebook. The K-means [1] and the LBG [2], count amongst the oldest of all VQ algorithms. The LBG mainly adopts a binary split approach that consists of splitting the centroids at each iteration, while partitioning the input space based on the centroids. The processing elements (PEs) are then updated such that they are placed at the centroids of all the partitions in the input space. Kohonen's SOM [3] is a stochastically and competitively trained vector quantizer. An important benefit of the SOM method is preservation of the topology of the input. This means, neighboring PEs in the weight space, correspond to neighboring points in the input (data) space. In summary, the SOM tries to approximate the distribution of the input data, while preserving structure.

One of the problems with the existing VQ algorithms is that they do not explicitly minimize a cost function; they are rather heuristic. Erwin [4] showed that when the SOM has converged, it is at the minimum of some discontinuous cost function. These discontinuities make the cost prone to drastic changes in some instances, which is undesirable. Heskes *et al.* [5] have made attempts to find a smooth cost function that, when minimized, gives the SOM update rule.

Efforts have also been made to design VQ algorithms using information theoretic perspectives. Heskes [5] used a cost function consisting of the quantization error and the entropy of the PEs. He also explored the links between SOM [3], elastic nets [6], and mixture modeling concluding that these methods are closely linked via the free energy point-of-view. Van Hulle [7] used a learning rule that consists of adapting the mean and variance of a Gaussian kernel, to maximize the entropy of the PEs. In order to prevent this algorithm from converging to a trivial solution where the PEs coincide, he modifies the algorithm quite heuristically to maximize entropy while minimizing mutual information by introducing competition between the kernels.

Earlier the authors approached the VQ problem from a density-matching point of view, where the statistical distributions of the data and the distribution of the PEs were matched through the maximization of the correlation, resulting in a cost function based on the Cauchy-Schwartz (CS) inequality [9]. In this paper, the VQ network weights are optimized to minimize the Kullback-Leibler (KL) divergence between the distribution of the data and the PEs. The equivalence between the minimization of KL divergence and the maximum likelihood principle is well known. Thus, the resulting optimal VQ solution can be considered equivalently as the maximum likelihood solution under the assumed distribution model. This algorithm based on KL divergence performs as well or better than the CS inequality algorithm, with reduced computational complexity.

Section II describes the proposed VQKL algorithm in detail. Section III presents simulation results using an artificial data set and a data set obtained by edge-detection of a face image. Comparisons with LBG and SOM are provided. The final section concludes the paper with remarks on possible future directions to improve the algorithm.

II. ALGORITHM

Consider the vector samples $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ from an information source in a *d*-dimensional signal space. Suppose that these samples are drawn from the distribution $g(\mathbf{x})$. Since, in practice the data distribution is generally unknown, it can be estimated using a Parzen-window estimator; this estimate of the data probability density function (pdf) is:

$$\hat{g}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{x} - \mathbf{x}_i; \mathbf{\Lambda}_{\mathbf{x}}) \quad (1)$$

where $K(\boldsymbol{\xi}; \mathbf{\Lambda})$ is the user-selected kernel function with $\mathbf{\Lambda}$ being the kernel size matrix and $\mathbf{x}_i$ are independent vector

samples drawn from the distribution $g(\mathbf{x})$. One of the requirements for the kernel function is that it should be symmetric, unimodal, and continuously differentiable. A Gaussian kernel meets all these requirements:

$$G(\boldsymbol{\xi};\boldsymbol{\Lambda}) = e^{-\boldsymbol{\xi}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}/2} \Big/ \left((2\pi)^{d/2} \mid \boldsymbol{\Lambda} \mid^{1/2}\right) \quad (2)$$

Similarly, let the true distribution of the PEs be $f(\mathbf{x})$. Suppose that the individual VQ weight vectors are independent samples drawn from this distribution, $\{\mathbf{w}_1,...,\mathbf{w}_M\}$. In VQ it is desirable to have $M<<N$. Using Parzen windowing with Gaussian kernels as before, the estimated density of the PEs is:

$$\hat{f}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} G(\mathbf{x}-\mathbf{w}_i, \boldsymbol{\Lambda}_{\mathbf{w}}) \quad (3)$$

The objective is to efficiently encode the data samples using a much smaller set of quantized weights without compromising the accuracy of the data representation. In other words, we wish to find a compact set of processing elements that can best represent the source data in terms of its distribution. This can be achieved by optimizing the weight vectors $\mathbf{w}_i$ such that the estimated density of the weights maximally match the estimated density of the data in accordance with some divergence criterion. Specifically, the Kullback-Leibler (KL) divergence [8], between two distributions $a(x)$ and $b(x)$, is

$$K(a \,||\, b) = \int a(x)\log\frac{a(x)}{b(x)}dx \qquad K(b \,||\, a) = \int b(x)\log\frac{b(x)}{a(x)}dx \quad (4)$$

All integrals are evaluated from -∞ to ∞. The KL divergence is not symmetric, i.e., $K(a \,||\, b) \neq K(b \,||\, a)$. Both quantities are nonnegative and become zero if and only if $a(x)=b(x)$.

A. Vector Quantization Using Kullback-Leibler Divergence

The VQKL algorithm uses the KL divergence measure as the optimality criterion. Due to the Parzen estimates of the densities using continuous and differentiable kernels, the performance surface is smooth, allowing us to use gradient-based or other iterative descent algorithms. In particular, the following cost function is minimized:

$$\begin{aligned} J(\mathbf{W}) &= \int f(\mathbf{x})\log\frac{\hat{f}(\mathbf{x})}{\hat{g}(\mathbf{x})}d\mathbf{x} \\ &= \int f(\mathbf{x})\log\hat{f}(\mathbf{x})d\mathbf{x} - \int f(\mathbf{x})\log\hat{g}(\mathbf{x})d\mathbf{x} \\ &= E_f[\log\hat{f}(\mathbf{x})] - E_f[\log\hat{g}(\mathbf{x})] \\ &\cong \frac{1}{M}\sum_{i=1}^{M}\log\frac{1}{M}\sum_{j=1}^{M} G(\mathbf{w}_i-\mathbf{w}_j;\boldsymbol{\Lambda}_{\mathbf{w}}) \\ &\quad -\frac{1}{M}\sum_{i=1}^{M}\log\frac{1}{N}\sum_{j=1}^{N} G(\mathbf{w}_i-\mathbf{x}_j;\boldsymbol{\Lambda}_{\mathbf{x}}) \end{aligned} \quad (5)$$

where $\mathbf{W}=[\mathbf{w}_1,\ldots,\mathbf{w}_M]$. The strategy of this cost function can be intuitively understood as follows: the first term is the negative of the Shannon entropy of the weights, therefore minimizing this cost is equivalent to maximizing the entropy of the weights (similar to [7]); at the same time, minimizing the second term in (5) can be considered as maximizing the correlation between the weight distribution $f(\mathbf{x})$ and log of the data distribution $\log(g(\mathbf{x}))$. The logarithm emphasizes the contributions from the low-probability regions of the data. This emphasis of sparse regions ensures that some weights are reserved for representing these areas in the data space. This cross term ensures that the weight distribution matches the data distribution closely.

The weight vectors are optimized by minimizing (5) using gradient descent:

$$\mathbf{w}_k(n+1) \leftarrow \mathbf{w}_k(n) - \eta\,\partial J(\mathbf{W})/\partial\mathbf{w}_k \quad (6)$$

The necessary gradient expressions with respect to each weight vector are found to be:

$$\begin{aligned} \frac{\partial J(\mathbf{W})}{\partial\mathbf{w}_k} &= \frac{-2}{M}\sum_{i=1}^{M}\frac{G(\mathbf{w}_i-\mathbf{w}_k;\boldsymbol{\Lambda}_{\mathbf{w}})\boldsymbol{\Lambda}_{\mathbf{w}}^{-1}(\mathbf{w}_i-\mathbf{w}_k)}{\rho(\mathbf{w}_i,\mathbf{w}_1,...,\mathbf{w}_M;\boldsymbol{\Lambda}_{\mathbf{w}})} \\ &\quad -\frac{1}{M}\frac{G(\mathbf{w}_k-\mathbf{x}_j;\boldsymbol{\Lambda}_{\mathbf{x}})\boldsymbol{\Lambda}_{\mathbf{x}}^{-1}(\mathbf{w}_k-\mathbf{x}_j)}{\rho(\mathbf{w}_k,\mathbf{x}_1,...,\mathbf{x}_N;\boldsymbol{\Lambda}_{\mathbf{x}})} \end{aligned} \quad (7)$$

where

$$\rho(\mathbf{w},\mathbf{x}_1,...,\mathbf{x}_N;\boldsymbol{\Lambda}) = \sum_{j=1}^{N} G(\mathbf{w}-\mathbf{x}_j;\boldsymbol{\Lambda}) \quad (8)$$

The alternative definition of KL divergence is not used because it reduces to only matching of the weight distribution to that of the data. This is easily seen by observing the explicit expression. The alternative divergence is:

$$\begin{aligned} J(\mathbf{W}) &= \int g(\mathbf{x})\log\frac{g(\mathbf{x})}{f(\mathbf{x})}d\mathbf{x} \\ &= \int g(\mathbf{x})\log g(\mathbf{x})d\mathbf{x} - \int g(\mathbf{x})\log f(\mathbf{x})d\mathbf{x} \end{aligned} \quad (9)$$

The first term does not depend on the weights, therefore it can be dropped from the cost function. Since the entropy maximization term is lacking, it has been observed that the convergence is typically much slower, although the computational load per iteration is lower in this alternative. Therefore, we adopt the approach given in (5) in the rest of the paper.

B. Discussion of Implementation Issues in VQKL

As in all gradient-based optimization techniques, this algorithm might suffer from local minima. It has been shown in previous papers that in learning algorithms designed using the Parzen windowing technique one way to avoid local minima is to anneal the kernel size [10]. A large kernel size will stretch and smoothen the performance surface eliminating some spurious local minima and enabling the PEs to move towards the *biased* global optimum of the new surface. As training progresses, the kernel size is annealed to yield a narrower kernel and a weaker smoothening effect, thus decreasing the bias in the global optimum allowing the weights to converge to the global optimum. Therefore, in the VQKL algorithm, we propose to start with a large kernel size to enable interactions between all PE-PE and PE-data pairs. By progressively annealing the kernel size with iterations, the interactions are limited to only nearby points. This

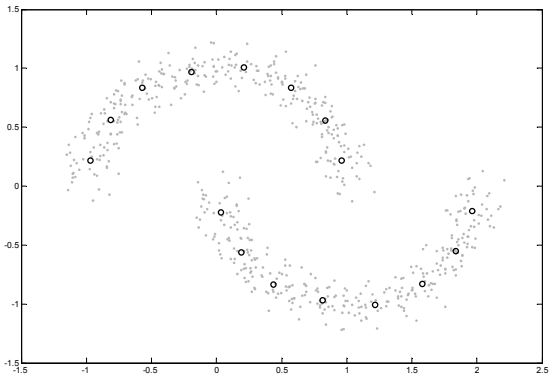

Fig. 1. Simulated data consisting of two half circles (dots). 16 PEs after convergence are shown in small circles.

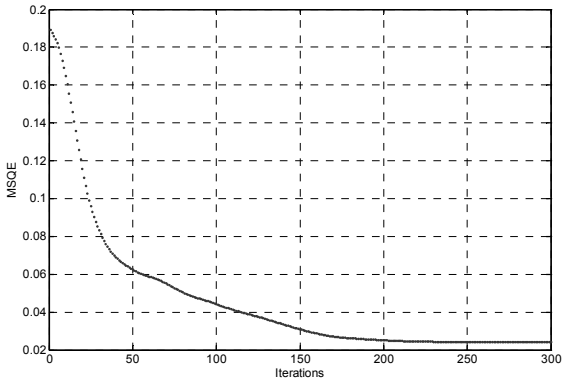

Fig. 2. Average MSQE versus iterations for VQKL in the first data set.

	VQKL	SOM	LBG
MSQE	0.024	0.024	0.023
J	2.437	3.378	2.460

Table 1. Comparison of MSQE for the three algorithms in the first data set. The standard deviations of VQKL and SOM are negligible over the Monte Carlo runs and for LBG it is zero.

progressive annealing strategy bears strong resemblance to the cooperative/competitive learning technique employed by the SOM.

Since VQKL uses batch updates, the kernel sizes are set up as follows:

a) Estimate the sample covariance matrix $\mathbf{\Sigma_x}$ of the data $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$.

b) Set $\mathbf{\Lambda_w}(0)=\mathbf{\Lambda_x}(0)=\alpha diag(\mathbf{\Sigma_x})$, where $\alpha>0$ is a constant determined empirically (typically in the order of 10 to 100), and $diag(\mathbf{\Sigma_x})$ is a diagonal matrix consisting of the variances of the data along each dimension.

c) Anneal both kernel sizes with every iteration (where n is the iteration index) using some annealing factor λ according to

$$\mathbf{\Lambda_w}(n)=\mathbf{\Lambda_x}(n)=\alpha diag(\mathbf{\Sigma_x})/(1+\lambda n) \quad (10)$$

The kernel size is never allowed to decrease below a selected threshold $\beta diag(\mathbf{\Sigma_x})$, where $\beta \geq 0$ is a small constant on the order of 10^{-3} to 10^{-1}.

The VQKL algorithm requires $O(M^2+MN)$ Gaussian evaluations for updating the weights at every iteration. The performance of the algorithm particularly depends on how accurately the densities are estimated using the Parzen window estimator. The kernel size matrices $\mathbf{\Lambda_w}$ and $\mathbf{\Lambda_x}$ constitute the free parameters of the density estimation process. Additionally, a gradient descent step size η must be selected. The step size must be sufficiently slow compared to the annealing rate. The step size can also be annealed to ensure smoother convergence.

III. Results

In this section, the quantization performance of the VQKL algorithm is demonstrated on two data sets. The first data set (also used in [9]) is an artificially generated two-cluster data in 2-dimensions. The second data set is an edge-detected face image, where the positions of the edge pixels in the image constitute the data points (also 2-dimensional). The second example is especially preferred as the edges of each organ in the face constitute a *natural* clustering solution. Comparisons with LBG and SOM are presented on these two data sets using standard performance metrics.

The first data set, shown in Fig. 1, essentially consists of samples drawn from two half circles with unit radius distorted with a Gaussian noise with standard deviation 0.1. Optimizing 16 randomly initialized PEs according to the KL divergence measure discussed above, the quantization solution shown in Fig. 1 is obtained consistently for all of the 20 Monte Carlo initializations. The average convergence curve of the algorithm over these Monte Carlo runs, quantified by the average mean-square-quantization-error (MSQE), is presented in Fig. 2. In this example, we set α=1.5, β=0, λ=0.08, the variances of the data in each direction were calculated as 0.75 and 0.51. The MSQE is calculated by:

$$\varepsilon=\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i-\mathbf{w}_{*i}\right\|^2 \quad (11)$$

where $\mathbf{w}_{*i}$ is the nearest weight to sample $\mathbf{x}_i$ after convergence is achieved. This is a widely accepted measure in the VQ literature and has become a standard error metric for performance evaluation. The MSQE of VQKL, LBG, and SOM are provided in Table 1 for the first data set. Since LBG explicitly tries to minimize this criterion, it performs the best among the three methods. Alternatively, distortion can also be quantified by the KL divergence, $J(\mathbf{W})$, between the source and the PE distribution (5). Even though $J(\mathbf{W})$ is explicitly used as the cost function in VQKL, it appears to be a stronger measure since it directly quantifies the extent to which the distribution of PEs differ from the distribution of the data. Evidently, higher order moments are considered in $J(\mathbf{W})$ as opposed to MSQE, which merely considers second

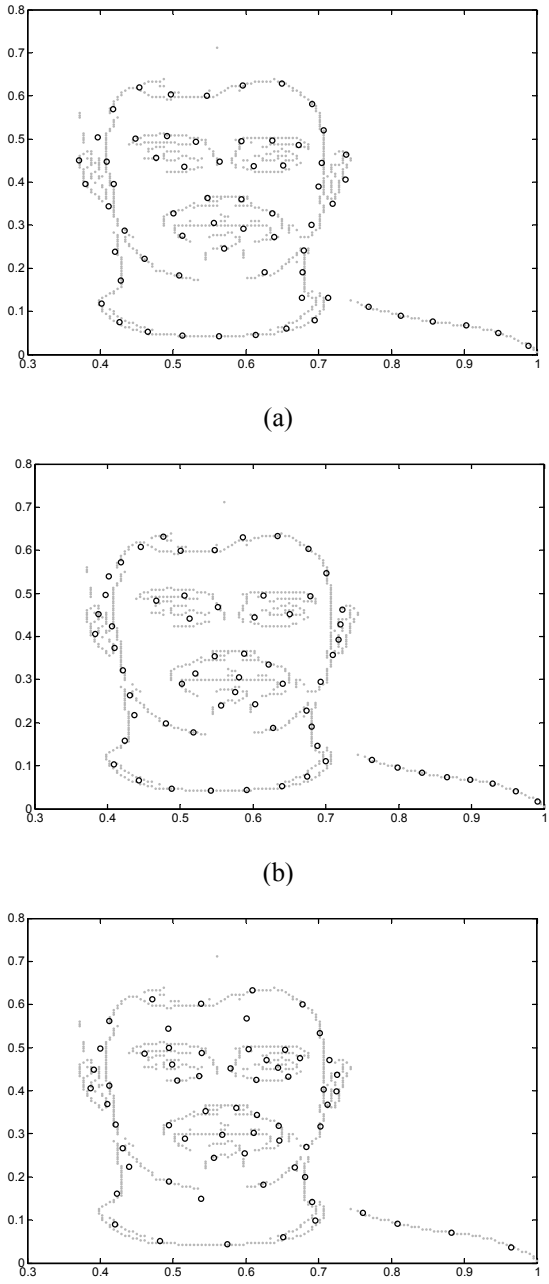

Fig 3. Illustration of the a) VQKL, b) LBG and c) SOM algorithms to quantize an edge-detected face image. 64 PEs are shown superimposed on the face data.

	VQKL	SOM	LBG
MSQE	3.37×10^{-4}	3.52e-4	3.06e-4
J	0.101	2.205	1.774

Table 2. Comparison of MSQE and KL divergence for the three algorithms in the face data set. The standard deviations of MSQE and J over the Monte Carlo runs are not provided as they were negligible.

order statistics. In this comparison, the VQKL outperforms both the SOM and the LBG by yielding the smallest KL divergence (also shown in Table 1).

The second example is the quantization of the edges of a face image. The weights are expected to specialize in interesting areas in the face, such as the ears, the nose, the eyes, and the mouth. This VQ representation of a face finds applications in face recognition and face modeling problems. A quantizer with 64 PEs is optimized on the image shown in Fig. 3a. Using the VQKL algorithm, the optimization results varied insignificantly over the 20 Monte Carlo runs performed with random initial conditions. The parameters were set to η=0.03, α=30, λ=0.12, and β=0. The data variances in each direction were found as 0.0171 and 0.0286. For the same image, the LBG quantization result is presented in Fig. 3b.

Even though the PE assignments in Fig. 3a and Fig. 3b look very similar, certain subtle qualitative differences are also evident. The left ear and the portion just above the right ear are described better by the VQKL compared to the LBG. The VQKL saves some weights from the shoulder representation to model the eyes with more precision, for example. This is expected because, intrinsically, the LBG tries to partition the regions and place the PEs at the centroids of the partitions, regardless of the distribution of the data. The VQKL on the other hand extracts more information from the data and allocates PEs to suit their structural properties. The bias in the LBG towards the centroids can also be seen on the shoulder region, in terms of the excessive number of PEs. For a quantitative comparison, the MSQE and $J(\mathbf{W})$ for VQKL, LBG, and SOM are provided in Table 2. As before, the LBG is better in terms of MSQE, while the VQKL outperforms the other two algorithms in terms of KL divergence.

IV. Discussion

In this study, we present an information theoretic approach to the vector-quantization problem. The proposed VQKL algorithm optimizes the code vectors by using the gradient descent technique to minimize the Kullback-Liebler (KL) divergence between the data distribution and the quantization weight vector distribution. As opposed to many existing VQ algorithms, which are based on heuristic reasonings, the VQKL algorithm is based on a well defined optimization problem, which also provides an intuitive notion of how the resulting VQ models the statistical distribution of the data. Its computational complexity is higher than that of the SOM and the LBG; however, the information extracted from the data enables a better infrastructure for quantization.

Comparisons on two data sets showed that the VQKL algorithm outperforms the other two in terms of quantization error entropy, which is a direct measure of quantization uncertainty according to information theory. In the future, the face image quantization example will be extended to the important application of face recognition. Other possible applications include speech recognition using the quantized features. Finally, the sensitivity of the least-squares type

optimality criterion to outliers is well known in the statistics and signal processing literature. The LBG method is expected to be heavily biased due to the strong effects of the outliers to the centroids. Since the outliers are defined as extremely rare cases of degenerate samples, the proposed method is expected to provide reduced sensitivity of the optimal VQ solution to outliers as they will not contribute significantly to the density mismatch between the data and the code vectors. The effects of outliers on the performance will be studied in detail in the future.

ACKNOWLEDGMENTS

This work was supported by NSF grant ECS-0300340.

REFERENCES

[1] S.P. Lloyd, "Least Squares Quantization in PCM's," Bell Telephone Laboratories Paper, Murray Hill, NJ, 1957.

[2] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Transactions on Communications, vol. 28, pp. 84-95, 1980.

[3] T. Kohonen, "Self-Organized Formation of Topologically Correct Features Maps," Biological Cybernetics, vol. 43, pp. 59-69, 1982.

[4] E. Erwin, K. Obermayer, K. Schulten, "Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions," Biological Cybernetics, vol. 67, pp. 47-55, 1992.

[5] T. Heskes, "Energy Functions for Self-Organizing Maps," in *Kohonen Maps*, E. Oja, S. Kaski (eds.), Elsevier, Amsterdam, pp. 303-316, 1999.

[6] R. Durbin, D. Willshaw, "An Analogue Approach of the Traveling Salesman Problem Using an Elastic Net Method," Nature, vol. 326, pp. 689-691, 1987.

[7] M.M. Van Hulle, "Kernel-Based Topographic Map Formation Achieved with an Information-Theoretic Approach", Neural Networks, vol. 15, pp. 1029-1039, 2002.

[8] S. Kullback, R.A. Liebler, "On Information and Sufficiency," The Annals of Mathematical Statistics, vol. 22, pp. 79-86, 1951.

[9] T. Lehn-Schioler, A. Hegde, D. Erdogmus, J.C. Principe, "Information Theoretic Vector-Quantization," submitted to Natural Computation, 2003.

[10] D. Erdogmus, J.C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," IEEE Transactions on Neural Networks, vol. 13, no. 5, pp. 1035-1044, 2002.