# Vector Quantization and Density Estimation

Robert M. Gray and Richard A. Olshen [*]
Information Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
http://www-isl.stanford.edu/~gray/compression.html

**Abstract**

The connection between compression and the estimation of probability distributions has long been known for the case of discrete alphabet sources and lossless coding. A universal lossless code which does a good job of compressing must implicitly also do a good job of modeling. In particular, with a collection of codebooks, one for each possible class or model, if codewords are chosen from among the ensemble of codebooks so as to minimize bit rate, then the codebook selected provides an implicit estimate of the underlying class. Less is known about the corresponding connections between lossy compression and continuous sources. Here we consider aspects of estimating conditional and unconditional densities in conjunction with Bayes-risk weighted vector quantization for joint compression and classification.

## 1 Introduction

The traditional goal of data compression in general and vector quantization in particular is to speed transmission or to minimize storage requirements of a signal while preserving the best possible quality of reproduction. This is usually formalized by trying to minimize the average distortion between the input and output in the sense of mean squared error (MSE) or a similar measure, subject to a constraint on the average bit rate. One approach to the design of such a system is to use a learning or training sequence of data, counting on the laws of large numbers to ensure that a code that does well for a training set will also do well for future data. Such designs implicitly or explicitly estimate the marginal distribution of the random process model for the signal source to be compressed. A common approach is to replace expectations used in theory by sample averages determined by the training set, essentially

[*] This work was supported by the National Science Foundation under NSF Grant No. MIP-931190 and by a gift from Hewlett-Packard, Inc. Corresponding author: R.M. Gray, rmgray@stanford.edu

using the empirical distribution of the training set as an estimate for the true, but unknown, distribution. This use of empirical distributions for source code design and their similar use in other statistical signal processing systems such as classification (detection) and regression (estimation) applications is both intuitive and useful for developing mathematical properties of such systems, especially those involving the convergence of algorithms and their performance in the limit of large training sets. That the empirical distribution is inherently discrete even if the original source is continuous does not usually cause any problems since the distribution estimate enters into the optimization only through expectations in a way that is not affected by the true distribution being a probability density function (pdf) or probability mass function (pmf).

The situation is different, however, in recent work aimed at incorporating the actual use of the compressed data in classification or regression by adding an additional cost function to the usual Lagrangian distortion measure involving MSE and bit rate. Here the Bayes risk, e.g., error probability, resulting from performing classification or regression on the compressed data is included in the overall distortion measuring the (lack of) quality of the compressed data. Optimal encoding of data described by a density function now requires an estimate of the complete density function and not just of expectations, which adds a new twist to traditional Lloyd-style code design algorithms. The goal here is to discuss the problem of density estimation for this particular context and several of the problems and approaches that arise. A specific "toy problem" studied by Kohonen et al. [1] is used to illustrate and compare some old and new approaches, including an old method with a new interpretation in terms of halftoning.

The basic problem considered is as follows. A training sequence $\{(x_n, y_n),\ i = 1, 2, \ldots, L\}$ which is a sample of a random process $\{(X_n, Y_n),\ i = 1, 2, \ldots\}$ is observed, and the individual $(X_n, Y_n)$ are assumed to have a common, but unknown, distribution $P_{XY}$ on a generic $(X, Y)$. Typically $P_X$ is absolutely continuous and is described by some pdf $f_X$ on $\Re^k$, and $P_Y$ is discrete, described by some pmf $p_Y$. It is desired to design, based on the training sequence, a vector quantizer for the observation $X$ which provides a good tradeoff among reproduction MSE, transmission or storage bit rate, and the Bayes risk resulting when $Y$ is guessed based on the encoded version of $X$. As will be seen, this basic problem leads to some issues simultaneously involving vector quantization and density estimation.

## 2 Bayes Vector Quantization

We begin by assuming that we are given a joint distribution $P_{XY}$ and summarize some of the ideas of Bayes risk weighted vector quantization (BVQ). Many details can be found, e.g., in [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], although here extensions to the variable rate case are considered in the basic development..

A BVQ consists of the following components:

**Encoder** $\tilde{\alpha} : A_X \to \mathcal{Z}$, where $A_X \subset \Re^k$ is the alphabet of the input vector and $\mathcal{Z}$ is the set of integers.

**Decoder** $\tilde{\beta} : \mathcal{Z} \to \hat{A}_X$, where $\hat{A}_X$ is the reproduction alphabet — often the same as $A_X$.

**Index Decoder** $\psi : \mathcal{Z} \to \mathcal{I} \subset \{0,1\}^*$, where $\mathcal{I}$ is a subset of the collection of all finite length binary sequences which satisfies the prefix condition. $\psi$ produces a binary channel codeword for transmission or storage. It is assumed to be a one-to-one invertible mapping of the range space of $\tilde{\alpha}$.

**Classifier** $\kappa : \mathcal{Z} \to A_Y$, where $A_Y$ is the alphabet of $Y$.

The decoder is also referred to as the "reproduction decoder" since its goal is to produce a vector resembling the original input vector. To measure the cost or error of inaccurate reproductions we assume a distortion measure $d(x, \hat{x}) \geq 0$. The most common example is the mean squared error

$$d(x, \hat{x}) = ||x - \hat{x}||^2 = \sum_{l=0}^{k-1} |x_l - \hat{x}_l|^2,$$

but most theory and algorithms also work for the more general input-weighted quadratic measures useful in perceptual coding [15, 16, 17, 18, 19, 20, 21] :

$$d(X, \hat{X}) = (X - \hat{X})^* B_X (X - \hat{X}),$$

$B_X$ positive definite. The average (ordinary) distortion resulting for a given code is $D(\tilde{\alpha}, \tilde{\beta}) = E[d(X, \tilde{\beta}(\tilde{\alpha}(X)))$.

The cost of the index decoder output is proportional to its length, the number of bits required to describe it. Given an $i \in \{0,1\}^*$, define the length function $l(i) =$ length of binary vector $i$. Define the *instantaneous rate* of a binary vector $i$ by

$$r(i) = \frac{i}{k}$$

in bits per input symbol. The *average rate* or *average codeword length* of the encoder applied to the source is defined by

$$R(\tilde{\alpha}, \psi) = E[r(\psi(\tilde{\alpha}(X)))].$$

The classifier is also a form of decoder, producing the receiver's best guess of the original, but not observed, $Y$. The cost of imperfect classification is measured by average Bayes risk:

$$B(\tilde{\alpha}, \kappa) \quad = \quad \sum_{k=1}^{M} \sum_{j=1}^{M} C_{j,k} \Pr(\kappa(\tilde{\alpha}(X)) = k \text{ and } Y = j),$$

where $C_{j,k}$ is cost of guessing $Y = k$ based on the encoded $X$ when the true class is $Y = j$. For simplicity we focus on the special case where $C_{j,k} = 1 - \delta_{k-j} = 0$ for

$k = j$ and 1 otherwise. In this case the average Bayes risk reduces to the probability of classification error.

A Langrangian cost function is formed in the manner of [22] to incorporate the separate costs of the three "decoders": MSE for the reproduction decoder, instantaneous rate for the index decoder, and Bayes risk for the classifier.

Given decoder $\tilde{\beta}$, index decoder $\psi$, and classifier $\kappa$, define the Lagrangian distortion between an input $x$ and an encoder output $i$ as

$$\begin{aligned}
\rho_{\lambda,\mu,P}(x,i) &= d(x,\tilde{\beta}(i)) + \mu r(\psi(i)) + \lambda \sum_{j=1}^{M} C_{j,\kappa(i)} P(Y=j|X=x) \\
J_{\lambda,\mu,P}(\tilde{\alpha},\tilde{\beta},\psi,\kappa) &= E[\rho_{\lambda,\mu,P}(X,\tilde{\alpha}(X))] \\
&= D(\tilde{\alpha},\tilde{\beta}) + \mu R(\tilde{\alpha},\psi) + \lambda B(\tilde{\alpha},\kappa)
\end{aligned}$$

The effect of the Lagrange multipliers can be illustrated by looking at the extreme points.

- $\mu \to \infty$ The cost in bits dominates and hence optimal code will have rate 0. Minimizing distortion and minimizing Bayes risk are incidental, so the best one can do at the decoder is $\min_v E[d(X,v)]$. This is accomplished by the single word codebook $\mathcal{C}_0 = \min_v^{-1} E[D(X,v)]$, the Lloyd centroid. Similarly, minimizing classification error is incidental, so the best one can do is pick the most probable class.
- $\mu \to 0$ Bits cost nothing relative to distortion and Bayes risk. Hence force 0 distortion and the classifier will be the minimum average Bayes risk classifier for the original source. This is only possible with a finite bit rate if the source is discrete with finite entropy, in which case the average bit rate is close to the entropy of the source vector in bits per coordinate.
- $\lambda \to \infty$ The emphasis is on classification, distortion and rate are incidental. Assuming the the allowed bit rate is large enough to losslessly code the classes (if they were known) at that bit rate, the strategy in this case is to use a cascade system that first applies optimal Bayes classifier to the source vector, then losslessly codes it so that the receiver will have the best possible classification. The distortion and bit rate are then minimized incidentally, given the class. This is simply a classified VQ [23, 24] as depicted in Figure 1 with a Bayes classifier, where the class codebooks are designed for the individual classes using an optimal bit allocation.
- $\lambda \to 0$ The emphasis is on compression, classification is incidental. This again suggests a cascade system: Ignore classification and design the best possible VQ in terms of trading off MSE and rate alone, then given the resuting code, label each codeword by the minimum average Bayes risk classification given that codeword. In other words, first compress and then classify. This, too, has a familiar form. The overall collection of reproduction codewords can be grouped by class label, resulting in a collection of codebooks, one for each class. This is just the structure of a universal source code or VQ (see, e.g., [25] and the references cited therein) as depicted in Figure 2, where the encoder constructs separate codebooks for different types of local behavior (here the class), and then finds the minimum distortion reproduction

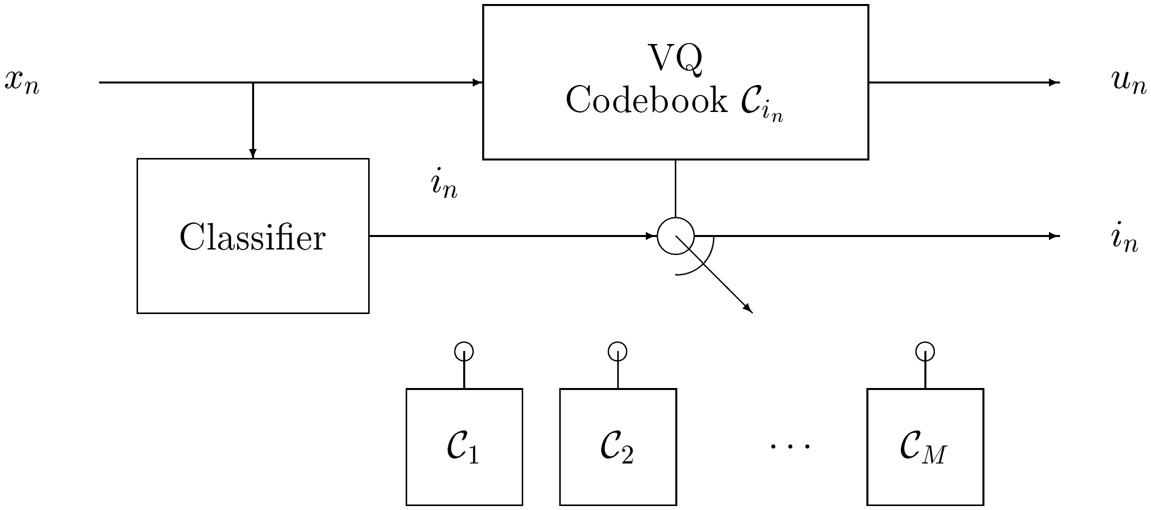

Figure 1: Classified VQ

(here combining MSE and rate) without regard to the accuracy of the resulting classification, which is nonetheless optimal for the given encoder.

# 3 Optimality Properties of Bayes VQ

Following the clustering (Lloyd) approach to quantizer design, we describe how to make each component optimal for the others and hence find necessary conditions for overall optimality. These properties in turn lead to a descent algorithm for designing the code.

The proofs are minor variations on the usual Lloyd optimality [24] (simple inequalities).

The components of the code are the encoder $\tilde{\alpha}$, the decoder $\tilde{\beta}$, the index decoder $\psi$, and the classifier $\kappa$. The goal is to minimize the Lagrangian distortion

$$J_{\lambda,\mu,P}(\tilde{\alpha},\tilde{\beta},\psi,\kappa) = D(\tilde{\alpha},\tilde{\beta}) + \mu R(\tilde{\alpha},\psi) + \lambda B(\tilde{\alpha},\kappa).$$

Note that for a given encoder $\tilde{\alpha}$, the remaining components can be independently optimized.

**Optimal Decoder** Given $\tilde{\alpha}$, $\psi$, $\kappa$, the optimal decoder is

$$\tilde{\beta}(i) = \min_{y\in\hat{A}}{}^{-1} E[d(X,y)|\tilde{\alpha}(X)=i],$$

the *Lloyd centroids* with respect to $P_X$.

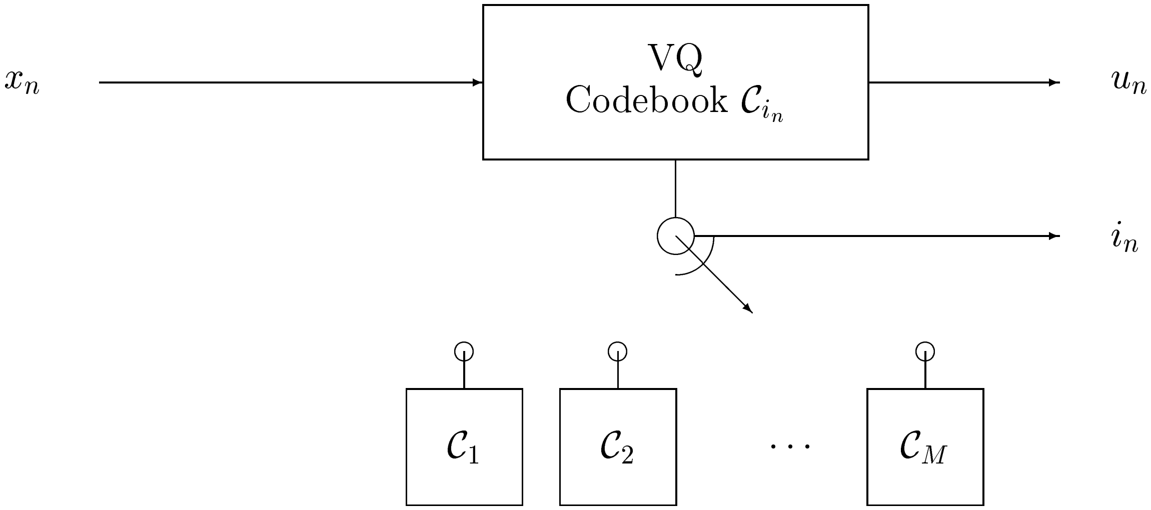

Figure 2: Universal VQ

In the MSE case, the centroids are the conditional means $E[X|\tilde{\alpha}(X) = i]$.

**Optimal Index Decoder** Given $\tilde{\alpha}$, $\tilde{\beta}$, $\kappa$, the optimal index decoder $\psi$ is the optimal lossless code for $\tilde{\alpha}(X)$, i.e., a Huffman code.

**Optimal Classifier** Given $\tilde{\alpha}$, $\psi$, $\tilde{\beta}$, the optimal classifier is

$$\kappa_{\text{Bayes}}(i) = \min_k{}^{-1}\{\sum_{j=1}^{M} C_{j,k}P(Y = j|\tilde{\alpha}(X) = i)\},$$

that is, the Bayes optimal classifier given the encoded input.

**Optimal Encoder** Given the $\kappa$, $\psi$, $\tilde{\beta}$, then the optimal encoder is

$$\begin{aligned}\tilde{\alpha}(x) &= \min_i{}^{-1}\rho_{\lambda,\mu,P}(x, i) \\ &= \min_i{}^{-1}\{d(x, \tilde{\beta}(i)) \\ &\quad +\mu r(\psi(i)) + \lambda\sum_{j=1}^{M} C_{j,\kappa(i)}P(Y = j|X = x)\}\end{aligned}$$

Iterating the three optimality properties provides a descent algorithm based on learning set (a generalized Lloyd algorithm) and pairwise application yields a tree-structured VQ (TSVQ).

The immediate and obvious drawback of the algorithm is that, unlike the usual Lloyd algorithm, the optimal encoder requires the class posterior probabilities $P(Y =$

$j|X = x)$. These must typically be estimated from the data. Unlike the distribution $P_X$ used in the usual Lloyd algorithm, these probabilities cannot be estimated simply by the empirical distribution implied by the training set. The empirical distribution defines these conditional probabilities *only for the $x$ contained in the training set*, yet the designed code will have to be well defined for all future $x$.

One obvious approach is to use a suboptimal encoder outside the training set, the reduced Lagrangian involving only MSE and bit rate. If the codes are assumed to be fixed rate, this reduces to a traditional nearest neighbor selection (yielding both compression and classification). Early examples of BVQ, however, showed that significant gains could be achieved in the parametric case of known distributions by using those distributions to do an optimal encoding. Furthermore, the encoder concentrating on compression and ignoring classification is effectively assuming the extreme of very small $\lambda$. This strategy is clearly suboptimal in the opposite extreme of large $\lambda$. This suggests that a better approach might be to estimate the class posterior probabilities from the training set and to then use these probabilities in the BVQ design as if they were the true probabilities.

# 4 BVQ and Density Estimation

The proposed strategy is to first design the estimator $\hat{P} = \{\hat{P}_{Y|X}(k|x),\ k \in \mathcal{H}; x \in A\}$ based on labeled learning set $\mathcal{L} = \{(x_n, y_n); n = 1, \ldots, L\}$. Then design $(\tilde{\alpha}, \tilde{\beta}, \kappa)$ using $\rho_{\lambda,\mu,\hat{P}}$. The two-step procedure yields a descent algorithm which should produce a good code if the density estimator is good.

The resulting encoder will operate in two steps: Given an input vector $x$,

- form $\hat{P}_{Y|X}(\cdot|x)$
- encode using optimum $\tilde{\alpha}$ for $\rho_{\lambda,\mu,\hat{P}}$.

Bayes' rule implies that the distribution estimation can be accomplished either by estimating pmfs or pdfs. In particular, if we the estimate class conditional pdfs $f_k(x) = f_{X|Y}(x|k)$ by $\hat{f}_k(x)$, $k \in A_Y$, then

$$\hat{P}_{Y|X}(y|x) = \frac{\hat{f}_k(x)\hat{P}_Y(k)}{\sum_m \hat{f}_m(x)\hat{P}_Y(m)},$$

where $\hat{P}_Y(k) =$ relative frequency of the class $k$ in $\mathcal{L}$

The intuition is that if the estimate is good, then the encoder should be close to optimal. The density estimate itself need not be sent to the decoder as it is not needed for decoding, only for encoding.

# 5 Design and Implementation

A variety of algorithms for designing Bayes vector quantization with posterior estimation have been developed and applied to artificial and real-world examples in the previously cited references. Examples include the development of methods for nonparametric estimation of the posterior class probabilities required for the Bayes VQ and studies of both tree-structured and full search Bayes VQs [13]. Specific techniques for estimating posterior probabilities include a tree-structured vector quantizer that is grown based on a relative entropy splitting rule [5] and two estimators [11] that are designed using the BFOS [42] variations on the CART$^{\mathrm{TM}}$ algorithm [43]. Specific examples include the combined compression and classification for segmentation of computerized tomographic images, aerial images, and mammograms [8, 7, 13, 11, 12]. To date the technique with the best performance (and, unfortunately, the highest computational complexity) uses CART-like algorithms to choose, transform, and discriminate among features [11].

Algorithms that provide joint compression and classification can be implemented with a hierarchical VQ (HVQ) [14]. HVQ is a table-lookup VQ that replaces the full search VQ encoder with a hierarchical arrangement of table lookups. With the combination of HVQ and a joint compression/classification design, the encoder, classifier and decoder can be implemented by table lookups so that there are no arithmetic computations required in the final implementation. The input vectors to the encoders can then be used directly as addresses in code tables to choose the codewords with the appropriate classification information. The system can thereby enable fast encoding, classifying, and decoding. It is useful for real-time video applications, such as, interactive video on the internet or video-teleconferencing. Preliminary results indicate that that for a reduction in encoding time of three-to-four orders of magnitude compared to standard full search or tree-structured VQ-based methods, the performance of the HVQ-based methods suffered only about 0.5 to 0.7 dB in compression and provided almost equivalent classification.

Several of the techniques developed for the combined goal of compression and classification can also be applied to classification of mixed-mode images, one example being color facsimile. There are advantages in being able to differentiate between and among different types of data. For example, the classification of educational videos into text and non-text blocks is useful for compression, since then different algorithms may be employed for the two types of data. Similar arguments can be made regarding the classification of document images into gradient shades, natural images, text, or background.

# 6 Density Estimation

Density estimation is a much studied topic in the statistical literature. See, e.g., Silverman [26] or Scott [27]. See also Devroye et al. [28] and Lugosi and Nobel [29] for theoretical aspects of vector quantization-based ("partition-based") density estimators. Well known approaches include kernel estimators, projection pursuit, tree-structured algorithms such as CART, and "partition-based" density estimators, which are effectively a VQ based estimator with the same form as the quantizer point densities of the Bennett-Gersho-Na-Neuhoff asymptotic quantization theory [30, 31, 32, 33, 34, 35, 36]. If a lattice quantizer is used, these can also be viewed as a form of "histogram density estimator," where the density is each Voronoi cell is the value of the histogram divided by the common cell volume.

The problem of density estimation is known to become particularly difficult with increasing vector dimension because of the famous "curse of dimensionality." In our case the problem is somewhat different from the traditional density estimation development. Although standard treatments often mention the application of densities to the classification problem, most end up by deciding that an estimator is good in a manner not directly connected to this application. For example, the most common goals are to show that with high probability the estimated pdf converges to the true pdf in $\mathcal{L}_p$ norm or in the relative entropy sense as the training set length grows. These results can be quite difficult, and they also appear to be more than is needed. In order to be useful in classification or in combined classification and compression, it is not necessarily true that the estimated density be a very close fit to the true one, only that it produce similar decision rules in the classification case or provide a similar weighting in the BVQ case.

In our application we effectively must estimate several densities, one class conditional density for each of the classes. The quality of the estimate will be measured by its overall effect on the combined compression and classification system. It is critically important that the estimator be simple, since it must be computed on the fly and used to encode each input vector.

We introduce a method of estimating class conditional probabilities for use in BVQ that is developed from an intuitive analogy with halftoning, but proves in implementation to be a simple implementation of a classical kernel method. The technique is explored using the Kohonen Gauss mixture example, where it is seen to provide the best performance to date on this simple source. The method is amenable to implementation using the the "fine-coarse" VQ approach of Moayeri and Neuhoff [37, 38], which in turn is a form of the hierarchical VQ of Chang et al. [39]. In particular, it is implementable by scalar quantizers followed by table lookups. The approach is also likely amenable to theory as the first stage can be evaluated using the Bennett asymptotic quantization theory for compression accuracy and the estimator accuracy evaluated using Vapnik-Cervonenkis theory [28, 44].

# 7 VQ PMF Estimation

Before discussing the proposed density estimator, we consider a natural approach to estimating class conditional probabilities that has been studied in the cited literature. Any VQ or TSVQ with MMSE (or other) encoder provides an estimate of the pmf $\hat{P}_{Y|X}$ via the relative frequencies of a training set. For example, if a quantizer $Q = (\tilde{\alpha}, \psi, \tilde{\beta})$ has Voronoi partition $\mathcal{S} = \{S_i;\ i = 1, \ldots, N\}$, $S_i = \{x : \tilde{\alpha}(x) = i\}$, then a conditional pmf estimate is the conditional relative frequency:

$$\begin{aligned}
\hat{P}_{Y|X}(k|x) &= \sum_i \hat{P}_{Y|\tilde{\alpha}(X)}(k|i) 1(x \in S_i) \\
&= \begin{cases} \sum_i \hat{P}_{Y|\tilde{\alpha}(X)}(k|i) & x \in S_i \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

where

$$\hat{P}_{Y|\tilde{\alpha}(X)}(k|i) = \frac{\sum_l 1(x_l \in S_i \text{ and } y_l = k)}{\sum_l 1(x_l \in S_i)}.$$

In a similar way, the input pdf for $X$ can be estimated by relative frequencies if the Bennett-style asymptotic assumption is made that the the Voronoi cells $S_i$ are sufficiently small to ensure that the pdf is roughly constant over each cell, so that the pdf can be approximated as the piecewise-constant function

$$\hat{f}_X(x) \approx \frac{\hat{P}_X(S_i)}{V(S_i)} 1(x_l \in S_i),$$

where

$$\hat{P}_X(S_i) = \frac{1}{L} \sum_l 1(x_l \in S_i)$$

and $V(S)$ is the volume of the cell, i.e., its Lebesgue measure. This provides a solution and is in fact just the classical partition-based pdf estimator or histogram pdf estimator. Unfortunately it is flawed for our application. First, piecewise densities have zero derivatives almost everywhere and no derivatives on the boundary, they are not continuous. This can yield poor models and cause problems in some applications. If the cells $S_i$ are relatively large, the estimator will be too coarse and will not accurately capture local variations in the pdf. On the other hand, if they cells are very small so that the Bennett approxiation can be trusted, there may not be enough training data to reliably estimate the cell probabilities and hence the unknown density. This latter problem is perhaps the most serious in designing a quantizer based on a training set. Lastly, the approach only makes sense when all the cells $S_i$ have finite volume.

All of these problems are eased if one uses a kernel pdf estimator. Instead of effectively adding a weighted indicator function at every sample point, one instead adds a smoothly varying function to produce an estimator that is continuous and not binary values. Instead of doing this only to estimate $f_X$, however, it can be done for the two class conditional pdf's $f_{X|Y}$, which can then be combined using Bayes' rule to form the overall pdf and the required conditionl pmf. The proposed technique turns out to do this, but it is introduced by a non-traditional argument.

# 8 Inverse Halftoning Density Estimation

The goal of density estimation is to convert a scatter plot of training data into a smooth estimate of the probability density that produced the data. Consider, for example, a 2-dimensional example given by the scatter plot of Figure 3 produced by a two-dimensional iid Gaussian source with 0 mean and variance 1. If the source is stationary and ergodic, then Glivenko-Cantelli arguments and the ergodic theorem dictate that the unknown pdf will be high where the spatial density of points is high, and low where it is low. This suggests an analogy with halftoning: suppose that it is desired to render a grayscale analog image into a digital halftoned image, i.e., a

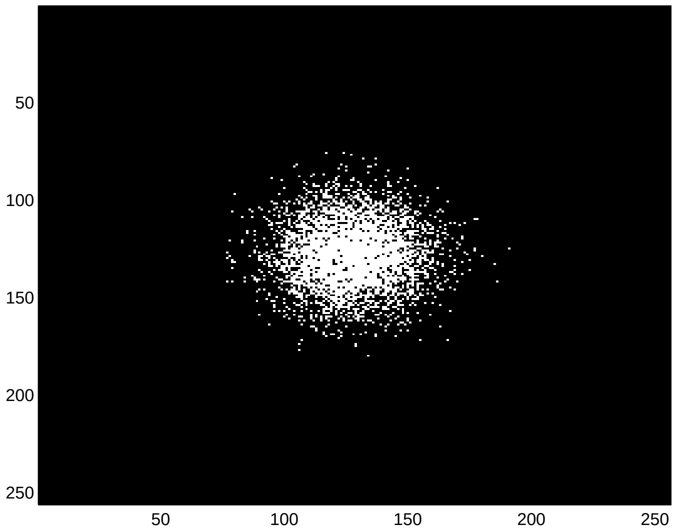

Figure 3: 2D Scatter Plot

raster image with pixels that can have only binary values. Halftoning techniques will produce an image such that where the intensity in the original is high, the density of unit intensity pixels will also be high, and where the intensity of the image is low, the density of unit intensity pixels will be low. The eye effectively performs a spatial low pass filtering on the resulting halftoned image, which then causes the binary rastered pixels to be perceived as a continuous tone image. By analogy the scatter plot as printed can be considered as the digital halftoned version of an original, unseen, continuous grayscale image, with the unknown pdf playing the role of the grayscale intensity. The scatter plot is "digital" in that that the plotting program (Matlab in our case) must discretize the axis and can plot the sample points only at these discrete locations. The analogy suggests that the original pdf can be recovered by inverse halftoning, that is, by passing the halftoned image (the scatter plot) through a spatial low pass filter. (In the halftoning literature, the halftoned image is often considered to be the original image plus *blue noise*, high pass noise to be removed by a low pass filter.)

The low pass filtering operation is standard basic signal processing: simply take an FFT of the 2-dimensional plot, weight the coefficients by the Fourier transform of a low pass filter, and then take the inverse FFT. Taking the magnitude of the resulting image ensures nonnegative intensities, which can then be normalized to provide a pdf (and a graphical display in the case of 2-dimensions). As to which low pass filter should be used, we here again invoke intuition to pick something fairly simple. An ideal low pass with a sharp cutoff would cause ringing in the reconstruction, which could produce unwanted artifacts. A common alternative is a simple, smooth Gaussian shape, which is what we chose. The bandwidth of the filter was chosen to be neither very narrow or very wide. In particular, we chose the exponential multiplier term in the Gaussian shape (1 over twice the variance if it were a pdf) to be $4/K$, where each dimension was quantized to $K$ levels. ($K$=128 or 256 in the example considered.)

If the digital aspect is ignored, then the inverse halftoning pdf estimator has a simple interpretation. The learning set can be considered as a sample density (or intensity) by putting a Dirac delta at each sample point. Taking the Fourier transform of the sum of the Dirac deltas produces just a sum of complex exponentials of equal weight. Each of these is then multiplied by the low pass filter transfer function $H$. Inverse transforming then yields a sum of shifted versions of the inverse transforms of $H$, that is, a sum of copies of the impulse response of the filter centered at each of the original sample point deltas. If $H$ is Gaussian, then so is its inverse transform, and hence the reconstruction is simply a sum of Gaussians centered at each training point. In other words, this is a classical kernel estimator with a Gaussian kernel. Although more complicated kernel methods may do better, e.g., by adapting the kernel shape to the local point density, the transform method has the advantage of simplicity and speed and should generalize to higher dimensions because of the separability of the transform. In summary, the proposed estimator is just a discretized version of a classical kernel estimator, accomplished by an FFT. The use of the FFT on a quantized version of the input for density estimation via Gaussian kernels was first proposed by Silverman [40] and subsequently studied in Jones and Lotwick [41] and Silverman [26], Section 3.5. This approach seems particularly well suited to the current application because of its implementability in hierarchical fashion. In particular, the FFT computations need not be done on line, but only when the pdf estimator is designed. Once designed, a table lookup suffices to compute the weighting for the distortion computation.

# 9 Kohonen's Example

As a particular "toy problem" that has been studied for classification, compression, and combinations of the two, we consider the two dimensional Gaussian mixture used by Kohonen et al. [1]. We consider random vectors $X = (X_0, X_1)$ described by a probability density function (pdf)

$$f_{X_0,X_1}(x_0,x_1) = \frac{1}{2} f^{(0)}_{X_0,X_1}(x_0,x_1) + \frac{1}{2} f^{(1)}_{X_0,X_1}(x_0,x_1),$$

where $f^{(k)}_{X_0,X_1}$ is the conditional pdf given that the class label $Y = k$ and $Y$ is equally likely to be 0 or 1. We also confine interest to the case of equal classification costs, so that the minimimum Bayes risk classifier with respect to the true distributions becomes a maximum a posteriori (MAP) classification rule:

$$\begin{aligned} \kappa_{\mathrm{MAP}}(x_0,x_1) &= \max_k{}^{-1} \Pr(Y = k | (X_0, X_1) = (x_0, x_1)) \\ &= \max_k{}^{-1} \frac{f^{(k)}_{X_0,X_1}(x_0,x_1)}{f^{(0)}_{X_0,X_1}(x_0,x_1) + f^{(1)}_{X_0,X_1}(x_0,x_1)}, \end{aligned} \qquad (1)$$

which can be explicitly solved for specific distributions and the resulting minimum Bayes risk computed. No VQ based classifier can provide lower Bayes risk.

In Kohonen's example, the two classes are each bivariate (two-dimensional) normal random vectors with iid components having 0 mean and equal variance. The variances are $\sigma_0^2 = 1$ and $\sigma_1^2 = 4$ for classes 0 and 1, respectively. The optimal encoder then needs to compute $P(Y = l|X = x) = f_{X|Y}(x|l)p_l / \sum_j f_{X|Y}(x|j)p_j = f_{X|Y}(x|l)/\sum_j f_{X|Y}(x|j)$, since the priors are equal, where $f_{X|Y}(x|l) = e^{-\frac{1}{2\sigma_l^2}||x||^2}/\sqrt{2\pi\sigma_l^2}$. Because equal costs are assumed, $C_{01} = C_{10} = 1$, and the optimal encoder, $\alpha^*(x)$, then becomes

$$\begin{aligned}
\alpha^*(x) &= \min_i{}^{-1}\left\{||x-\beta(i)||^2 + \lambda\left[\frac{1(\{\kappa(i)=1\})e^{-\frac{1}{2}||x||^2} + 1(\{\kappa(i)=0\})\frac{1}{2}e^{-\frac{1}{8}||x||^2}}{e^{-\frac{1}{2}||x||^2}+\frac{1}{2}e^{-\frac{1}{8}||x||^2}}\right]\right\} \\
&= \min_i{}^{-1}\left\{||x-\beta(i)||^2 + \lambda\left[\frac{1(\{\kappa(i)=1\})e^{-\frac{3}{8}||x||^2} + 1(\{\kappa(i)=0\})\frac{1}{2}}{e^{-\frac{3}{8}||x||^2}+\frac{1}{2}}\right]\right\} \qquad (2)
\end{aligned}$$

Note that the two distributions overlap, and thus there is no decision rule providing perfect classification. The Bayes decision rule, which minimizes the Bayes risk (or classification error since we have equal costs), is a circle about the origin of radius 1.923. This optimal decision rule yields an error probability of .264..

The mean squared error can be bound using the Shannon lower bound by

$$D \geq \frac{e^{-2(R-\frac{1}{4}[\ln(2\pi e)+\ln(2\pi e4])}}{2\pi e} = \sqrt{2\pi e}4^{\frac{1}{4}}e^{-2R}. \qquad (3)$$

In the case of the scatter plot of figure 3, the inverse halftone estimator results in the pdf of figure 4 and the resulting overall estimate of $P_{Y|X}(0|x)$. The estimator

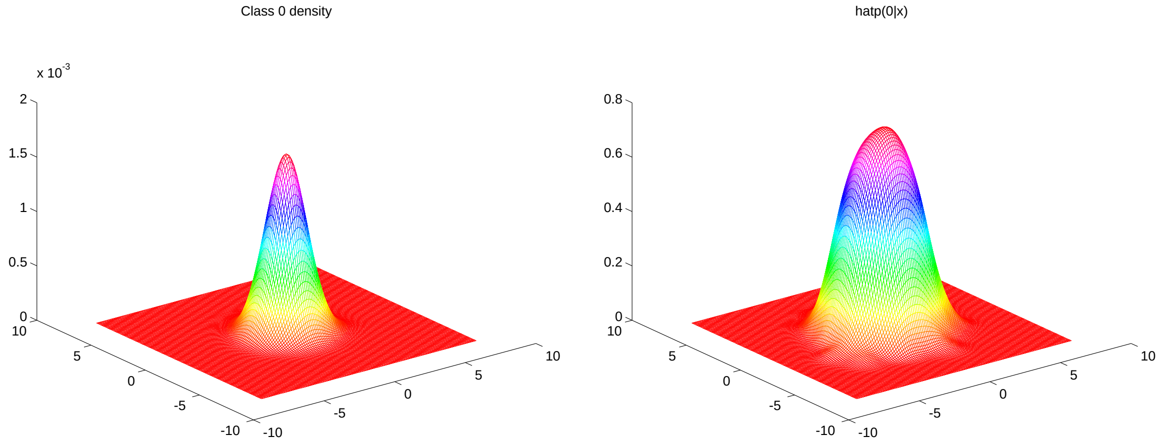

Figure 4: Inverse halftone pdf estimate (left) and conditional pmf estimate (right)

is seen to be smooth and provides the estimate necessary to design the complete Bayes VQ. One way of visually judging the estimators quality is to use it simply as

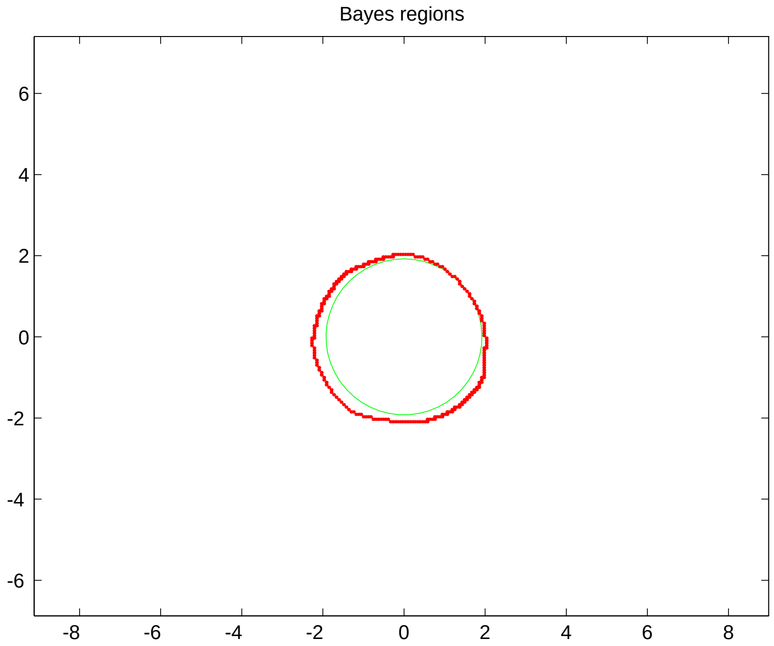

Figure 5: True and estimated Bayes classification region

a classifier without regard to compression. If the estimated pdf is substituted for the true pdf in a minimum average Bayes risk classifier, then the result can be depicted as in figure 5. The actual minimum Bayes risk (or minimum a posteriori in our example) classifier simply tests to see whether or not the received vector is inside or outside of a circle of radius 1.923 centered at the origin. The estimated pdf results in the approximation to the circle shown.

In [4, 10, 12], 5 trials were run, each using a training set of 10,000 vectors to design the code, and a separate test set of 10,000 vectors to test the code. We expanded this to 10 trials for the new density estimator. To make fair comparisons with earlier results (including Kohonen), we used fixed rate codes, i.e., we did not constrain the entropy. We chose a different value of $\lambda$, however, as the cited BVQ used $\lambda = 100$ while we used $\lambda = 10$. As will be seen, both were high enough to effectively place full emphasis on classification.

The test results for the 10 trials were then averaged to form an overall measure of algorithm performance. For the basic experiment, each design trial was from scratch: for each training sequence a new pdf estimator was designed and an initial codebook was designed as a simple minimum MSE VQ. The initial codebook and the pdf estimator were then used to initiate the BVQ design. During design the Bayes risk term in distortion measure only required a table lookup for each input vector to get the bias to the MSE in the Langrangian distortion computation. The Bayes Lloyd iteration typically converged very fast, within four or so iterations. The performance for each test and the overall average are given in Table 1, along with results using Kohonen's "learning VQ" for classification modified to improve compression [13] and

| | | |
|---|---|---|
| Trial 1 | .610 | .266 |
| Trial 2 | .621 | .268 |
| Trial 3 | .598 | .270 |
| Trial 4 | .608 | .272 |
| Trial 5 | .632 | .264 |
| Trial 6 | .664 | .262 |
| Trial 7 | .614 | .271 |
| Trial 8 | .656 | .257 |
| Trial 9 | .647 | .258 |
| Trial 10 | .614 | .268 |
| Average | .626 | 0.265 |
| Cascade | .610 | .299 |
| Kohonen LVQ | .725 | .279 |
| MSE Enc BVQ | .594 | .289 |
| TSVQ pmf estimator | .630 | .270 |
| Parametric BVQ | .620 | .264 |

Table 1: MSE and $P_e$ for Kohonen's Example

the results for a parametric BVQ, i.e., BVQ design with known pdfs. Also shown are a cascade system with minimum MSE VQ followed by a Bayes classifier given the coded output, and a BVQ designed for the training set using the "omniscient" class conditional probability estimator (the relative frequency of the class for the given training vector value) with a suboptimal minimum MSE encoder outside the training set. The proposed algorithm uniformly outperforms Kohonen's LVQ [45, 1] in both classification and compression performance. Also presented for comparison is the performance of a cascade code using an MSE VQ followed by a Bayes classifier given the coded input. This provides the best possible MSE, but the classifier, even though optimized for the quantizer output, performs notably worse than the joint design. The TSVQ pmf estimating two step coder [12] uses a tree-structured VQ to estimate the class pmf's conditioned on the input, without explicitly doing density estimation. The tree was grown by splitting the node with the largest partial Bayes distortion. Once the estimating tree was grown, a full search BVQ was designed using the resulting estimator. Note that the inverse halftoning estimator provides better performance in terms of both squared error and probability of error, even though the Lagrangian multiplier is much smaller. Of perhaps most importance is the fact that the average classifier performance for the density estimating BVQ is extremely close to that achieved by the parametric BVQ where the density is known, and both are extremely close to the optimal classifier performance *based on the original, unquantized, inputs.* Thus for only a slight increase in MSE, the jointly designed codebook provides a significant improvement in Bayes risk. Alternatively, this suggests that for the given value of $\lambda$, the jointly designed system acts like a classified VQ, first performing a Bayes classification and then doing an optimum MSE VQ for each class, implicitly optimizing the bit allocation between classes. This suggests a conjecture that for $\lambda$ large enough, the joint structure reduces to a cascade of optimal classification followed by optimal compression for the chosen class.

Our final figures further illustrate the example. The Voronoi diagram for the parametric case [4, 10] where the the pdf's are actually known is shown in figure 6. Here the inner cells coincide exactly with the actual Bayes region. The MSE Voronoi regions resulting from the same codebook is also shown. Here the polygonal regions do not well approximate the circle and the Bayes risk increases. The Voronoi cells for the two-step BVQ with estimation using a TSVQ pmf estimator [12] is shown in figure 7. Here It can be seen that the cells form a better approximation to the circle than do the MSE encoded regions and the Bayes risk falls, but they are still visibly jagged.

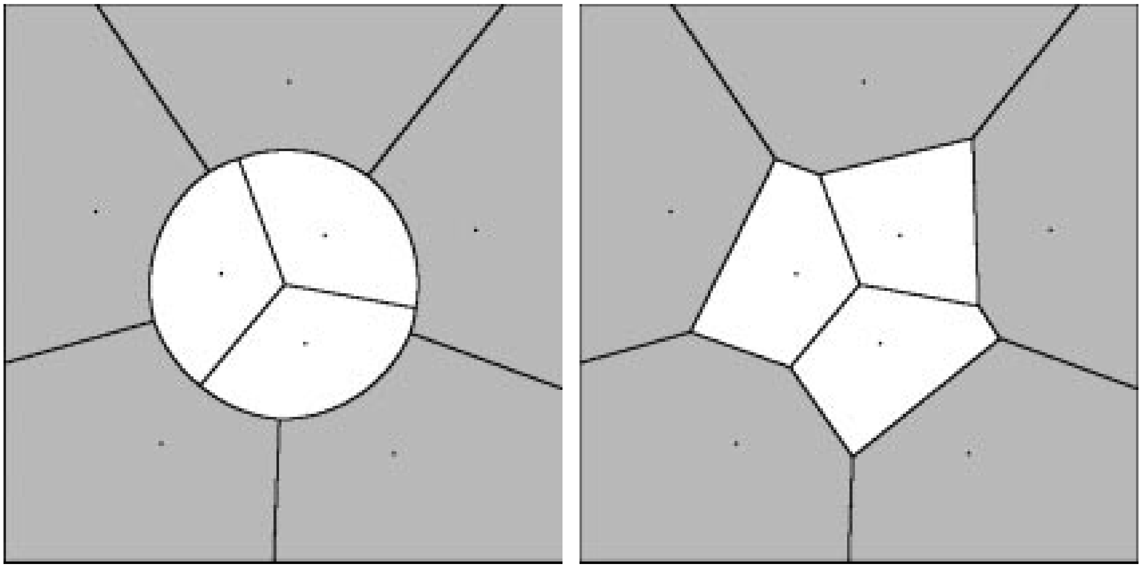

Figure 6: Parametric BVQ Voronoi diagram (left) and MSE encoded Voronoi diagram

Figure 8 shows the result of the Lloyd improvement algorithm on an initial code. The initial codewords are denoted by ‘o’s, the interim codewords resulting at the end of each iteration by ‘+’s, and the final codewords by ‘*’s. The resulting Voronoi cells with respect to the Lagrangian distortion measure are plotted in figure 9. The figure is plotted by finding the borders of the Voronoi cells on the same “fine quantized” lattice used to produce the pdf estimates. The Voronoi diagram resulting from an MSE encoder is also shown. Note the MSE distortion requires planar (here affine) boundaries to the cells, while those of the modified distortion appear curved and provide a better fit to the circle.

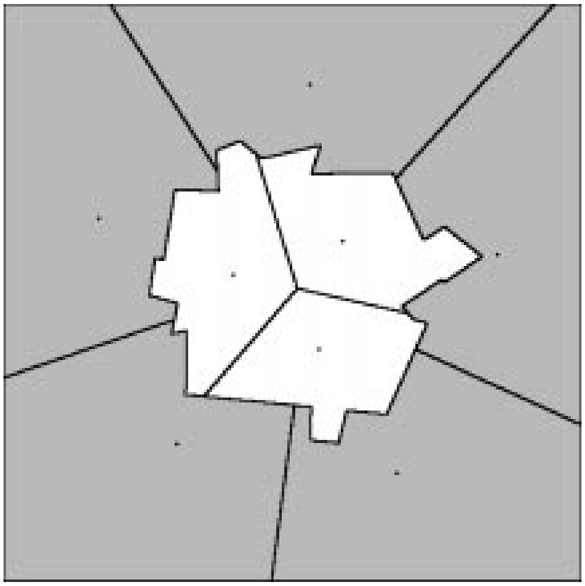

Figure 7: TSVQ pdf Estimating BVQ

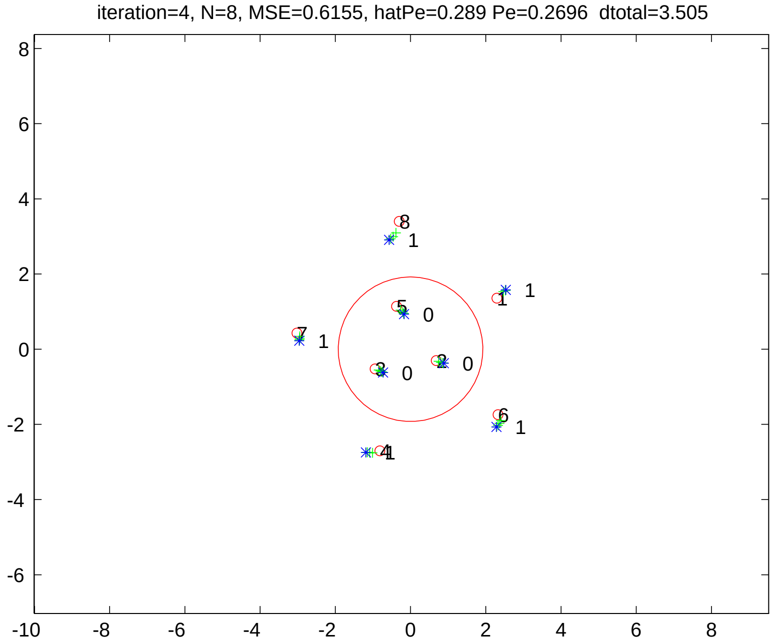

Figure 8: BVQ Design Iterations

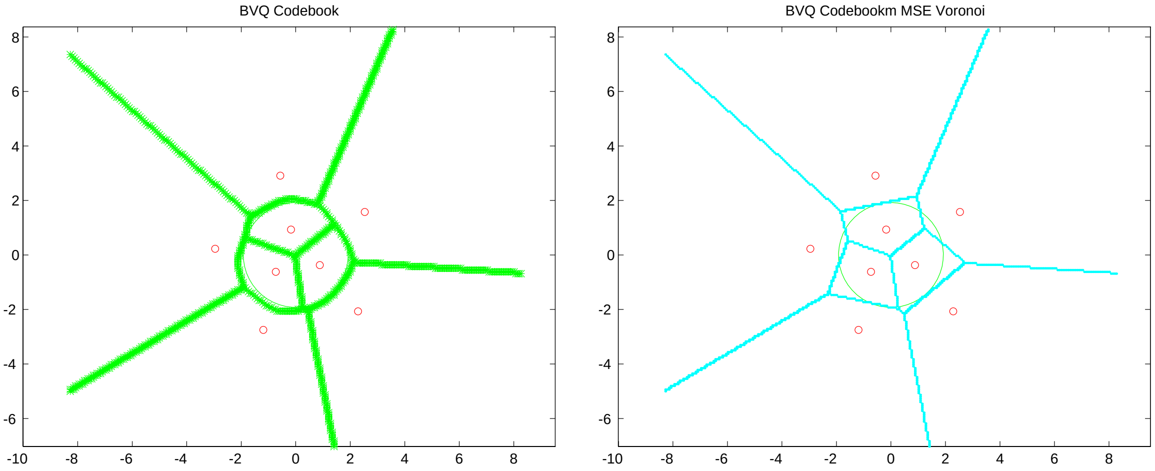

Figure 9: BVQ Voronoi regions (left) and MSE encoded regions (right)

# 10 Conclusions

The problem of designing a code that both compresses and classifies has been described and related to the problem of estimating unknown probability densities from labeled training sets. A simple density estimation technique motivated by inverse halftoning and based on low pass filtering via an FFT was introduced and shown to provide excellent performance on a toy problem. In particular it was shown to outperform Kohonen's popular LVQ and a cascade of optimal MSE quantization followed by optimal classification. For the Lagrange multiplier chosen, the system essentially provided classification as good as optimal Bayes classification on the original vector, while providing only slightly suboptimal MSE at 1.5 bits per symbol.

The technique raises many interesting theoretical problems and work is beginning towards its extension to real image data and its comparison with other methods for classification and combined compression and classification.

*Acknowledgements*

The authors acknowledge the many helpful comments of students and colleagues in the Compression and Classification Group of Stanford University

`http://www-isl.stanford.edu/~gray/compression.html`

Several of the quoted previous results were due to Group alumnae Karen Oehler and Keren Perlmutter.

# References

[1] T. Kohonen, G. Barna, and R. Chrisley, "Statistical pattern recognition with neural networks: benchmarking studies," in *IEEE International Conference on Neural Networks*, pp. I–61–68, July 1988.

[2] K. Oehler and R. Gray, "Combining image classification and image compression using vector quantization," in *Proceedings of the 1993 IEEE Data Compression Conference (DCC)*, J. Storer and M. Cohn, eds., Snowbird, Utah, pp. 2–11, IEEE Computer Society Press, March 1993.

[3] R.M. Gray, K.L. Oehler, K.O. Perlmutter, and R.A. Olshen, "Combining tree-structured vector quantization with classification and regression trees," *Proceedings of the Twenty-seventh Asilomar conference on Signals, Systems, & Computers*, 31 October – 3 November, 1993, Pacific Grove, CA, pp. 1494–1498.

[4] K.L. Oehler, *Image Compression and Classification using Vector Quantization*, Ph.D. Dissertation, Stanford University, 1993.

[5] K.O. Perlmutter, R.M. Gray, K.L. Oehler, R.A. Olshen, "Bayes risk weighted tree-structured vector quantization with posterior estimation," *Proceedings Data Compression Conference (DCC)*, IEEE Computer Society Press, March 1994, pp. 274–283.

[6] R.D. Wesel and R.M. Gray, "Bayes risk weighted VQ and learning VQ," *Ibid.*, pp. 400–409.

[7] C. L. Nash, K. O. Perlmutter, and R. M. Gray. "Evaluation of bayes risk weighted vector quantization with posterior estimation in the detection of lesions in digitized mammograms," In *Proceedings of the 28th Asilomar Conference on Circuits Systems and computers*, Pacific Grove, CA, October 1994, p. 716-20 vol.1.

[8] K. O. Perlmutter, C. L. Nash, and R. M. Gray. "A comparison of Bayes risk weighted vector quantization with posterior estimation with other VQ-based classifiers." In *Proceedings of the IEEE 1994 International Conference on Image Processing* (ICIP), volume 2, pages 217–221, Austin, TX, Nov. 1994.

[9] K.O. Perlmutter, C.L. Nash, and R.M. Gray, " Bayes Risk Weighted Tree-structured Vector Quantization with Posterior Estimation," *Ibid.*, Volume 2, pp. 217-221 November 1994.

[10] K.L. Oehler and R.M. Gray, "Combining image compression and classification using vector quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pp. 461–473, May 1995.

[11] K. O. Perlmutter, R. M. Gray, R. A. Olshen, S. M. Perlmutter, "Bayes Risk Weighted Vector Quantization with CART Estimated Posteriors," *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing* (ICASSP), pp. 2435-8 vol.4, May 1995.

[12] K. O. Perlmutter, "Compression and Classification of Images using Vector Quantization and Decision Trees," Ph.D. Thesis, December 1995.

[13] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes Risk Weighted Vector Quantization with Posterior Estimation for Image Compression and Classification," *IEEE Transactions on Image Processing*, vol.5, no.2, p. 347-60, February 1996.

[14] Chaddha, N., K. Perlmutter and R.M. Gray, "Joint image classification and compression using hierarchical table-lookup vector quantization." Proceedings of Data Compression Conference - DCC '96. Held: Snowbird, UT, USA, 31 March-3 April 1996. (USA: IEEE Comput. Soc. Press, 1996. p. 23-32)

[15] N.B. Nill and B.H. Bouxas, "Objective image quality measure derived from digital image power spectra," *Optical Engineering*, Vol 31, pp. 813–825, April 1992.

[16] S. Daly, "Visible differences predictor: an algorithm for the assessment of image fidelity," *SPIE Proceedings*, Vol 1666, pp. 2–14, 1992.

[17] R. M. Gray, P. C. Cosman, and K. Oehler, "Incorporating visual factors into vector quantization for image compression," contributed chapter in *Digital Images and Human Vision*, B. Watson, Ed., MIT Press, Cambridge, Mass, 1993, pp. 35–52.

[18] W. R. Gardner and B. D. Rao, "Theoretical Analysis of the High-Rate Vector Quantization of LPC Parameters," *IEEE Transactions on Speech and Audio Processing*, Vol 3, pp. 367-381, September 1995.

[19] A. M. Eskicioglu and P. S. Fisher, "Image Quality Measures and Their Performance," *IEEE Transactions on Communications*, Vol 43, pp. 2959-2965, December, 1995.

[20] R.M. Gray and E. Karnin, "Multiple local optima in vector quantizers," *IEEE Transactions on Information Theory*, Vol. 28, pp. 708–721, November 1981.

[21] J. Li, N. Chaddha, and R.M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," 1997 IEEE International Symposium on Information Theory, Ulm, Germany, June 1997. (Full paper submitted for possible publication. Preprint available at `http://www-isl.stanford.edu/~gray/compression.html`.)

[22] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust. Speech & Signal Proc.*, Vol. 37, pp. 31–42, January 1989.

[23] A. Gersho and B. Ramamurthi, "Image coding using vector quantization," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, (Paris), pp. 428–431, April 1982.

[24] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Press, 1992.

[25] P.A. Chou, M. Effros, and R.M. Gray, "A vector quantization approach to universal noiseless coding and quantization," *IEEE Trans. on Information Theory*, Vol. 42, No. 4, July 1996, pp. 1109–1138.

[26] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Monographs on Statistics and Applied Probability 26, Chapman & Hall, New York, fourth printing, 1994.

[27] D.W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley, New York, 1992.

[28] L. Devroye, L. Györfi, and G. Lugosi, *A Probabalistic Theory of Pattern Recognition*, Springer, New York, 1996.

[29] G. Lugosi and A. Nobel, "Consistency of data-driven histogram methods for density estimation and classification," *Ann. Statist.* **24**, April 1996, 687–706.

[30] W.R. Bennett, "Spectra of quantized signals," *Bell Systems Technical Journal*, Vol. 27, pp. 446–472, July 1948.

[31] P.L. Zador, "Topics in the asymptotic quantization of continuous random variables," unpublished Bell Laboratories Memorandum, 1963.

[32] P.L. Zador, "Development and evaluation of procedures for quantizing multivariate distributions," Ph.D. Dissertation, Stanford University, 1963.

[33] P.L. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, Vol. 28, pp. 139–148. March 1982.

[34] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on Information Theory*, Vol. 25, pp. 373–380. July 1979.

[35] Y. Yamada, S. Tazaki, and R.M. Gray, "Asymptotic performance of block quantizers with a difference distortion measure," *IEEE Transactions on Information Theory*, Vol. 26, pp. 6–14. March 1980.

[36] S. Na and D.L. Neuhoff, 'Bennett's integral for vector quantizers," *IEEE Trans on Inform Thy*, Vol. 41, pp 886-900, July 1995.

[37] N. Moayeri, D.L. Neuhoff, and W.E. Stark, "Fine-coarse vector quantization," *IEEE Transactions on Signal Processing*, Vol. 39, pp. 1503–15, July 1991.

[38] N. Moayeri and D.L. Neuhoff, "Theory of lattice-based fine-coarse vector quantization," *IEEE Transactions on Information Theory*, Vol. 37, pp 1072–84, July 1991.

[39] P.C. Chang, J. May, and R.M. Gray, "Hierarchical vector quantizers with table-lookup encoders," Globecom, June 1985.

[40] "Algorithm AS176.kernel density estimation using the fast Fourier transform," *Appl. Statist.*, Vol. 31, 93–99.

[41] M.C. Jones and H.W. Lotwick, "A remark on Algorithm AS176. Kernel density estimation using the fast Fourier Transform,' Remark AS R50, *Appl. Statist.*, Vol. 33, 120–122, 1984.

[42] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory,* Vol. IT-35, pp. 299-315, March 1989.

[43] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Belmont,California: Wadsworth, 1984.

[44] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1996.

[45] T. Kohonen, *Self-organization and associative memory.* Berlin: Springer-Verlag, third ed., 1989.

[46] K. O. Perlmutter, N. Chaddha, J. Buckheit, R. A. Olshen, and R. M. Gray, "Text Segmentation in Mixed Mode Images using Classification Trees and Transform Tree-Structured Vector Quantization," *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, May 1996, 2231–2234.