Scalar and Vector Quantization

Mário A. T. Figueiredo,
Departamento de Engenharia Electrotécnica e de Computadores,
Instituto Superior Técnico, Lisboa, Portugal

mario.figueiredo@ist.utl.pt

November 2008

Quantization is the process of mapping a continuous or discrete scalar or vector, produced by a source, into a set of digital symbols that can be transmitted or stored using a finite number of bits. In the case of continuous sources (with values in $\mathbb{R}$ or $\mathbb{R}^n$) quantization must necessarily be used if the output of the source is to be communicated over a digital channel. In this case, it is, in general, impossible to exactly reproduce the original source output, so we're in the context of lossy coding/compression.

In this lecture notes, we will review the main concepts and results of scalar and vector quantization. For more details, see the book by Gersho and Gray [2], the accessible tutorial by Gray [3], or the comprehensive review by Gray and Neuhoff [4]

1 Scalar Quantization

1.1 Introduction and Definitions

Let us begin by considering the case of a real-valued (scalar) memoryless source. Such a source is modeled as a real-valued random variable, thus fully characterized by a probability density function (pdf) f_X. Recall that a pdf f_X satisfies the following properties: $f_X(x) \geq 0$, for any $x \in \mathbb{R}$,

$$\int_{-\infty}^{\infty} f_X(x)\, dx = 1,$$

and

$$\int_{a}^{b} f_X(x)\, dx = P[X \in [a, b]],$$

where $P[X \in [a, b]]$ denotes the probability that the random variable X takes values in the interval $[a, b]$. To avoid technical issues, in this text we only consider continuous pdfs.

Consider the objective of transmitting a sample x of the source X over a binary channel that can only carry R bits, each time it is used. That is, we can only use R bits to encode each sample of X. Naturally, this restriction implies that we are forced to encoding any outcome of X into one of $N = 2^R$ different symbols (binary words). Of course, this can be easily generalized for D-ary channels (instead of binary), for which the number of different words is D^R; however, to keep the notation simple, and without loss of generality, we will only consider the case of binary channels. Having received one of the $N = 2^R$ possible words, the receiver/decoder has to do its best to recover/approximate the original sample x, and it does so by outputting one of a set of N values.

The procedure described in the previous paragraph can be formalized as follows. The encoder is a function $\mathcal{E} : \mathbb{R} \to \mathcal{I}$, where $\mathcal{I} = \{0, 1, ..., N-1\}$ is the set of possible binary words that can be sent through the channel to represent the original sample x. Since the set $\mathcal{I}$ is *much smaller* than $\mathbb{R}$, this function is non-injective and there are many different values of the argument that produce the same value of the function; each of this sets is called a quantization region, and is defined as

$$R_i = \{x \in \mathbb{R} : \mathcal{E}(x) = i\}.$$

Since $\mathcal{E}$ is a function defined over all $\mathbb{R}$, this definition implies that the collection of quantization regions (also called cells) $\mathcal{R} = \{R_0, ..., R_{N-1}\}$ defines a *partition* of $\mathbb{R}$, that is,

$$(i \neq j) \Rightarrow R_i \cap R_j = \emptyset \qquad \text{and} \qquad \bigcup_{i=0}^{N-1} R_i = \mathbb{R}. \tag{1}$$

The decoder is a real-valued function $\mathcal{D} : \mathcal{I} \to \mathbb{R}$; notice that since the argument of $\mathcal{D}$ only takes N different values, and $\mathcal{D}$ is a deterministic function, it can also only take N different values, thus its range is a finite set $\mathcal{C} = \{y_0, ..., y_{N-1}\} \subset \mathbb{R}$. The set $\mathcal{C}$ is usually called the *codebook*. The i-th element of the codebook, y_i, is sometimes called the *representative* of the region/cell R_i.

Considering that there are no errors in the channel, the sample x is reproduced by the decoder as $\mathcal{D}(\mathcal{E}(x))$, that is, the result of first encoding and then decoding x. The composition of the functions $\mathcal{E}$ and $\mathcal{D}$ defines the so-called *quantization function* $\mathcal{Q} : \mathbb{R} \to \mathcal{C}$, where $\mathcal{Q}(x) = \mathcal{D}(\mathcal{E}(x))$. The quantization function has the following obvious property

$$(x \in R_i) \Leftrightarrow \mathcal{Q}(x) = y_i, \tag{2}$$

which justifies the term *quantization*. In other other, any x belonging to region R_i is *represented* at the output of the system by the corresponding y_i.

A quantizer (equivalently a pair encoder/decoder) is completely defined by the set of regions $\mathcal{R} = \{R_0, ..., R_N\}$ and the corresponding representatives $\mathcal{C} = \{y_0, ..., y_{N-1}\} \subset \mathbb{R}$. If all the cells are intervals (for example, $R_i = [a_i, b_i[$ or $R_i = [a_i, \infty[$) that contain the corresponding representative, that is, such that $y_i \in R_i$, the quantizer is called *regular*. A regular quantizer in which all the regions have the same length (except two of them, which may be unbounded on the left and the right) is called a *uniform* quantizer. For example, the following set of regions

and codebook define a 2-bit ($R = 2$, thus $N = 4$) regular (but not uniform) quantizer:

$$\mathcal{R} = \{]-\infty, -0.3],]-0.3, 1.5],]1.5, 4[, [4, \infty[\} \qquad \text{and} \qquad \mathcal{C} = \{-1, 0, 2, 5\}.$$

As another example, the following set of regions/cells and codebook define a 3-bit ($R = 3$, thus $N = 8$) uniform quantizer:

$$\begin{aligned} \mathcal{R} &= \{]-\infty, 0.3],]0.3, 1.3],]1.3, 2.3],]2.3, 3.3],]3.3, 4.3],]4.3, 5.3],]5.3, 6.3],]6.3, \infty[\} \\ \mathcal{C} &= \{0, 1, 2, 3, 4, 5, 6, 7\}. \end{aligned}$$

1.2 Optimal Quantizers, Lloyd's Algorithm, and the Linde-Buzo-Gray Algorithm

1.2.1 Expected Distortion

Finding an optimal scalar quantizer consists in finding the set of regions, $\mathcal{R}$, and the codebook, $\mathcal{C}$, minimizing a given objective function, which measures quantizer performance. Although there are other possibilities, the standard quantity used to assess the performance of a quantizer is the *expected distortion*

$$E\left[d(X, \mathcal{Q}(X))\right] = \int_{-\infty}^{\infty} f_X(x)\, d\left(x, \mathcal{Q}(x)\right)\, dx,$$

where $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a so-called *distortion measure*. Among the several reasonable choices for d, such as $d(x, z) = |x - z|$, the one which is, by far, most commonly used, is the squared error, $d(x, z) = (x - z)^2$. With the squared error, the expected distortion becomes the well-known *mean squared error* (MSE),

$$\text{MSE} = E\left[(X - \mathcal{Q}(X))^2\right] = \int_{-\infty}^{\infty} f_X(x)\, \left(x - \mathcal{Q}(x)\right)^2\, dx.$$

The MSE is also called the quantization noise power.

1.2.2 Optimal Quantizer

Adopting the MSE to measure the quantizer performance, the problem of finding the optimal set of regions and corresponding representatives becomes

$$\left(\mathcal{R}^{\text{opt}}, \mathcal{C}^{\text{opt}}\right) = \arg\min_{\mathcal{R}, \mathcal{C}} \int_{-\infty}^{\infty} f_X(x)\, \left(x - \mathcal{Q}(x)\right)^2\, dx, \tag{3}$$

under the constraint that the regions that constitute $\mathcal{R}$ have to satisfy the condition in (1).

Because the set of regions constitutes a partition of $\mathbb{R}$ (see (1)), and because of (2), the integral defining the MSE can be written as

$$\text{MSE}(R_0, ..., R_{N-1}, y_0, ..., y_{N-1}) = \sum_{i=0}^{N-1} \int_{R_i} f_X(x)\, (x - y_i)^2\, dx, \tag{4}$$

where the notation $\text{MSE}(R_0, ..., R_{N-1}, y_0, ..., y_{N-1})$ is used to emphasize that the mean squared error depends on the quantization regions, $R_0, ..., R_{N-1}$ and their representatives $y_0, ..., y_{N-1}$.

1.2.3 Partial Solutions

It is, in general, extremely hard to find the global minimizer of $\text{MSE}(R_0, ..., R_{N-1}, y_0, ..., y_{N-1})$, simultaneously with respect to all the regions and representatives. However, it's possible to solve two partial problems:

- Given the quantization regions regions $\mathcal{R} = \{R_0, ..., R_{N-1}\}$, find the corresponding optimal codebook,

$$\{y_0^*, ..., y_{N-1}^*\} = \arg \min_{y_0^*, ..., y_{N-1}} \sum_{i=0}^{N-1} \int_{R_i} f_X(x) \, (x - y_i)^2 \, dx. \tag{5}$$

- Given a codebook $\mathcal{C} = \{y_0, ..., y_{N-1}\}$, find the optimal regions,

$$\begin{aligned} \{R_0^*, ..., R_{N-1}^*\} &= \arg \min_{R_0^*, ..., R_{N-1}} \int_{-\infty}^{\infty} f_X(x) \, (x - \mathcal{Q}(x))^2 \, dx & (6) \\ &\text{subject to} \quad (i \neq j) \Rightarrow R_i \cap R_j & (7) \\ &\qquad \bigcup_{i=0}^{N-1} R_i = \mathbb{R}. & (8) \end{aligned}$$

Let us start by examining (5); observe that the function being minimized is the sum of N non-negative functions, and each of these functions only depends on one element of $\mathcal{C}$. Consequently, the problem can be solved independently with respect to each y_i, that is,

$$y_i^* = \arg \min_y \int_{R_i} f_X(x) \, (x - y)^2 \, dx.$$

Expanding the square in the integrand leads to

$$\begin{aligned} y_i^* &= \arg \min_y \left[\int_{R_i} f_X(x) \, x^2 \, dx + y^2 \int_{R_i} f_X(x) \, dx - 2 \, y \int_{R_i} f_X(x) \, x \, dx \right] & (9) \\ &= \arg \min_y \left[y^2 \int_{R_i} f_X(x) \, dx - 2 \, y \int_{R_i} f_X(x) \, x \, dx \right], & (10) \end{aligned}$$

where the second equality is due to the fact that the first term in the right hand side of (9) does not depend on y, thus it is irrelevant for the minimization. The minimum is found by computing the derivative with respect to y, which is

$$\frac{d}{d\,y} \left[y^2 \int_{R_i} f_X(x) \, dx - 2 \, y \int_{R_i} f_X(x) \, x \, dx \right] = 2 \, y \int_{R_i} f_X(x) \, dx - 2 \int_{R_i} f_X(x) \, x \, dx$$

and equating it to zero, which leads to the following equation

$$y \int_{R_i} f_X(x) \, dx = \int_{R_i} f_X(x) \, x \, dx,$$

the solution of which is

$$y_i^* = \frac{\int_{R_i} f_X(x)\, x\, dx}{\int_{R_i} f_X(x)\, dx}. \tag{11}$$

This expression for the optimal representative of region R_i has a clear probabilistic meaning. Observe that the conditional density of X, conditioned by the event $A_i = (X \in R_i)$ is, according to Bayes law,

$$f_{X|A_i}(x|A_i) = \frac{f_{X,A_i}(x, A_i)}{P[A_i]} = \frac{P[A_i|x] f_X(x)}{P[A_i]} = \frac{1_{R_i}(x) f_X(x)}{P[X \in R_i]},$$

where $1_{R_i}(x) = 1$, if $x \in R_i$, and $1_{R_i}(x) = 0$, if $x \notin R_i$, is called the *indicator function* of region R_i. Computing the expected value of X, conditioned by the event that $A_i = (X \in R_i)$,

$$\begin{aligned} E[X|X \in R_i] &= \int_{-\infty}^{\infty} x\, f_{X|A_i}(x|A_i)\, dx \\ &= \frac{1}{P[X \in R_i]} \int_{-\infty}^{\infty} x\, f_X(x)\, 1_{R_i}(x)\, dx \\ &= \frac{\int_{R_i} x\, f_X(x)\, dx}{\int_{R_i} f_X(x)\ dx}, \end{aligned} \tag{12}$$

which is exactly expression (11). This shows that the optimal representative of the cell R_i is the conditional expected value of the random variable X, given that X is in R_i. A more *physical* interpretation of (11) is that it is the center of (probabilistic) mass of region R_i.

Let us now examine problem (6)–(8), where we seek the optimal regions, given a codebook $\mathcal{C} = \{y_0, ..., y_{N-1}\}$. Notice that the fact that there is no restriction on the form of the regions R_i (apart from those in (7) and (8)) means that choosing the regions is the same as selecting, for each x, what is its "best" representative among the given $\{y_0, ..., y_{N-1}\}$. In mathematical terms, this can be written as the following inequality

$$\int_{-\infty}^{\infty} f_X(x)\, (x - \mathcal{Q}(x))^2\, dx \geq \int_{-\infty}^{\infty} f_X(x)\, \min_i (x - y_i)^2\, dx;$$

that is, since the codebook $\{y_0, ..., y_{N-1}\}$ is fixed, the best possible encoder is one that chooses, for each x, the closest representative. In conclusion, the optimal regions are given by

$$R_i = \{x :\, (x - y_i)^2 \leq (x - y_j)^2,\, j \neq i\}, \quad \text{for} \quad i = 0, ..., N-1, \tag{13}$$

that is, R_i is the set of points that are closer to y_i than to any other element of the codebook.

1.2.4 The Lloyd Algorithm

The Lloyd algorithm for quantizer design works by iterating between the two partial solutions described above.

Step 1: Given the current codebook $\mathcal{C}^{(t)} = \left\{y_0^{(t)}, ..., y_{N-1}^{(t)}\right\}$, obtain the optimal regions

$$R_i^{(t)} = \{x : (x - y_i^{(t)})^2 \leq (x - y_j^{(t)})^2,\ j \neq i\}, \quad \text{for } i = 0, ..., N-1;$$

Step 2: Given the current regions $\mathcal{R}^{(t)} = \left\{R_0^{(t)}, ..., R_{N-1}^{(t)}\right\}$, update the representatives

$$y_i^{(t+1)} = \frac{\displaystyle\int_{R_i^{(t)}} f_X(x)\, x\, dx}{\displaystyle\int_{R_i^{(t)}} f_X(x)\, dx}, \quad \text{for } i = 0, ..., N-1;$$

Step 3: Check some stopping criterion; if it is satisfied, stop; if not, set $t \leftarrow t+1$, and go back to Step 1.

A typical stopping criterion would be to check if the maximum difference between two consecutive values of codebook elements is less than some threshold; that is, the algorithm would be stopped if the following condition is satisfied

$$\max_i (y_i^{(t)} - y_i^{(t+1)})^2 \leq \varepsilon. \tag{14}$$

Under certain conditions, Lloyd's algorithm converges to the global solution of the optimization problem (3); however, these conditions are not trivial and way beyond the scope of these lecture notes. In fact, the convergence properties of the Lloyd algorithm are a topic of current active research; the interested reader may look at the recent paper by Du, Emelianenko, and Ju [1].

1.2.5 Zero Mean Quantization Error of Lloyd Quantizers

Algorithms obtained by the Lloyd algorithm satisfy simultaneously the partial optimality conditions (11) and (13) and are called Lloyd quantizers. These quantizers have the important property that the expected value of the quantization error is zero, that is, $E\left[\mathcal{Q}(X) - X\right] = 0$, or, equivalently, $E[\mathcal{Q}(X)] = E[X]$. To show this, we write

$$\begin{aligned}
E\left[\mathcal{Q}(X)\right] &= \int_{-\infty}^{\infty} f_x(x)\, \mathcal{Q}(x)\, dx && (15)\\
&= \sum_{i=0}^{N-1} y_i \int_{R_i} f_x(x)\, dx && (16)\\
&= \sum_{i=0}^{N-1} \left(\frac{\displaystyle\int_{R_i} x\, f_x(x)\, dx}{\displaystyle\int_{R_i} f_x(x)\, dx} \right) \int_{R_i} f_x(x)\, dx && (17)\\
&= \sum_{i=0}^{N-1} \int_{R_i} x f_x(x)\, dx && (18)\\
&= \int_{-\infty}^{\infty} x f_x(x)\, dx && (19)\\
&= E[X]. && (20)
\end{aligned}$$

1.2.6 The Linde-Buzo-Gray Algorithm

Very frequently, instead of knowledge of the pdf of the source, $f_X(x)$, what we have available is a set of samples $\mathcal{X} = \{x_1, ..., x_n\}$, where n is usually (desirably) a large number. In this scenario, the optimal quantizer will have to be obtained (learned) from these samples. This is what is achieved by the Linde-Buzo-Gray (LBG) algorithm, which is a sample version of the Lloyd algorithm. The algorithm is defined as follows.

Step 1: Given the current codebook $\mathcal{C}^{(t)} = \left\{y_0^{(t)}, ..., y_{N-1}^{(t)}\right\}$, obtain the optimal regions

$$R_j^{(t)} = \{x : (x - y_j^{(t)})^2 \leq (x - y_k^{(t)})^2, \, k \neq j\}, \quad \text{for } \; j = 0, ..., N-1;$$

Step 2: Given the current regions $\mathcal{R}^{(t)} = \left\{R_0^{(t)}, ..., R_{N-1}^{(t)}\right\}$, update the representatives

$$y_j^{(t+1)} = \frac{\sum\limits_{i:\, x_i \in R_j^{(t)}} x_i}{n_j^{(t)}}, \quad \text{for } \; j = 0, ..., N-1,$$

where $n_j^{(t)} = |\mathcal{X} \cap R_j^{(t)}|$ is the number of samples in $R_j^{(t)}$.

Step 3: Check some stopping criterion; if it is satisfied, stop; if not, set $t \leftarrow t+1$, and go back to Step 1.

As in the Lloyd algorithm, a typical stopping criterion has the form (36).

Notice that we don't need to explicitly define the regions, but simply to assign each point to one of the current regions $\left\{R_0^{(t)}, ..., R_{N-1}^{(t)}\right\}$. That is, the Step 1 of the LBG algorithm can be written with the help of indicator variables w_{ij}, for $i = 1, ..., n$, and $j = 0, ..., N-1$, defined as follows:

$$w_{ij} = 1 \Leftrightarrow j = \arg\min_k \left\{ \left(x_i - y_k^{(t)}\right)^2, \; k = 1, ..., N \right\},$$

that is w_{ij} equals one if and only if x_i is closer to y_j than to any other other element of the current codebook; otherwise, it is zero. With these indicator variables, the Step 2 of the LBG algorithm can be written as

$$y_j^{(t+1)} = \frac{\sum\limits_{i=1}^{n} x_i \, w_{ij}}{\sum\limits_{i=1}^{n} w_{ij}}, \quad \text{for } \; i = 0, ..., N-1,$$

that is, the updated j-th element of the codebook is simply the mean of all the samples that currently are in region $R_i^{(t)}$.

1.3 High-Resolution Approximation

Although there is an algorithm to design scalar quantizers, given the probability density function of the source (Lloyd's algorithm), or a set of samples (LBG algorithms), the most commonly used quantizers are uniform and of high resolution (large N). It is thus important to be able to have a good estimate of the performance of such quantizers, which is possible using the so-called "high-resolution approximation".

1.3.1 Uniform Quantizers

In uniform quantizers, all the regions R_i are intervals with the same width, denoted Δ. Of course, if X is unbounded (for example, $X \in \mathbb{R}$ and Gaussian) it is not possible to cover $\mathbb{R}$ with a finite number of cells of finite width Δ. However, we assume that we have enough cells to cover the region of of $\mathbb{R}$ where $f_X(x)$ is not arbitrarily close to zero. For example, if $X \in \mathbb{R}$ and $f_X(x)$ is a Gaussian density of zero mean and variance σ^2, we may consider that X is essentially always in the interval $[-4\,\sigma,\, 4\,\sigma]$, since the probability that X belongs to this interval is 0.9999.

The high-resolution approximation assumes that Δ is small enough so that $f_X(x)$ is approximately constant inside each quantization region. Under this assumption, the optimal representative for each region is its central point, thus $R_i = [y_i - \Delta/2,\, y_i + \Delta/2[$, and the MSE is given by

$$\begin{aligned} \text{MSE} &= \sum_{i=0}^{N-1} \int_{y_i-\Delta/2}^{y_i+\Delta/2} f_X(x)\,(x-y_i)^2\,dx \\ &\simeq \sum_{i=0}^{N-1} f_X(y_i) \int_{y_i-\Delta/2}^{y_i+\Delta/2} (x-y_i)^2\,dx \\ &= \sum_{i=0}^{N-1} f_X(y_i)\Delta \int_{y_i-\Delta/2}^{y_i+\Delta/2} \frac{1}{\Delta}\,(x-y_i)^2\,dx. \end{aligned} \tag{21}$$

Making the change of variables $z_i = x - y_i$ in each of the integrals, they all become equal to

$$\int_{y_i-\Delta/2}^{y_i+\Delta/2} \frac{1}{\Delta}\,(x-y_i)^2\,dx = \int_{-\Delta/2}^{\Delta/2} \frac{z^2}{\Delta}\,dz = \frac{\Delta^2}{12};$$

inserting this result in (21), and observing that $f_X(y_i)\Delta \simeq P[X \in R_i] \equiv p_i$ we obtain

$$\text{MSE} \simeq \frac{\Delta^2}{12} \sum_{i=0}^{N-1} p_i = \frac{\Delta^2}{12}, \tag{22}$$

since $\sum_i p_i = 1$.

If the width of the (effective) support of $f_X(x)$ is, say A, the number of cells N is given by $N = A/\Delta$. Recalling that $N = 2^R$, we have

$$\text{MSE} = \frac{A^2\; 2^{-2\,R}}{12}, \tag{23}$$

showing that each additional bit in the rate R produces an MSE reduction by a factor of 4. In terms of signal to (quantization) noise ratio (SNR), we have

$$\text{SNR} = 10 \log_{10} \frac{\sigma^2}{\text{MSE}} \, dB$$

where σ^2 denotes the source variance. Using the expression above for the MSE, we have

$$\text{SNR} = \underbrace{10 \log_{10} \frac{\sigma^2 \, 12}{A^2}}_{K} + R \, \underbrace{20 \log_{10} 2}_{\simeq 6.0} \simeq (K + 6\,R) \; dB,$$

showing that each extra bit in the quantizer achieves an improvement of approximately $6\,dB$ in the quantization SNR.

Notice that all the results in this subsection are independent of the particular features (such as the shape) of the pdf $f_X(x)$.

1.3.2 Non-uniform Quantizers

In non-uniform high-resolution quantizers, the width of each cell R_i is Δ_i, but it is still assumed that Δ_i is small enough so that $f_X(x)$ is essentially constant inside the cell R_i. Under this assumption, the optimal representative for region R_i is its central point, thus we can write $R_i = [y_i - \Delta_i/2, \, y_i + \Delta_i/2[$, and the MSE is given by

$$\begin{aligned}
\text{MSE} &= \sum_{i=0}^{N-1} \int_{y_i - \Delta_i/2}^{y_i + \Delta_i/2} f_X(x) \, (x - y_i)^2 \, dx \\
&\simeq \sum_{i=0}^{N-1} f_X(y_i) \int_{y_i - \Delta_i/2}^{y_i + \Delta_i/2} (x - y_i)^2 \, dx \\
&= \sum_{i=0}^{N-1} f_X(y_i) \Delta_i \int_{y_i - \Delta_i/2}^{y_i + \Delta_i/2} \frac{1}{\Delta_i} \, (x - y_i)^2 \, dx. \qquad (24)
\end{aligned}$$

Making the change of variables $z_i = x - y_i$ in each integral leads to

$$\int_{y_i - \Delta_i/2}^{y_i + \Delta_i/2} \frac{1}{\Delta_i} \, (x - y_i)^2 \, dx = \int_{-\Delta_i/2}^{\Delta_i/2} \frac{z^2}{\Delta_i} \, dz = \frac{\Delta_i^2}{12};$$

inserting this result in (24), and observing that $f_X(y_i)\Delta_i \simeq P[X \in R_i] \equiv p_i$ we obtain

$$\text{MSE} \simeq \sum_{i=0}^{N-1} p_i \, \frac{\Delta_i^2}{12}.$$

naturally, (22) is a particular case of the previous expression, for $\Delta_i = \Delta$.

1.4 Entropy of the Output of a Scalar Encoder

The output of the encoder, $I = \mathcal{E}(X)$, can be seen as a discrete memoryless source, producing symbols from the alphabet $\mathcal{I} = \{0, 1, ..., N-1\}$, with probabilities

$$p_i = P[X \in R_i] = \int_{R_i} f_X(x)\, dx, \quad \text{for } i = 0, 1, ..., N-1.$$

The entropy of $\mathcal{E}(X)$ provides a good estimate of the minimum number of bits required to encode the output of the encoder, and (as will be seen below) will provide a coding theoretical interpretation to the differential entropy of the source X.

The entropy of I is given by

$$H(I) = -\sum_{i=0}^{N-1} p_i \log p_i = -\sum_{i=0}^{N-1} \left(\int_{R_i} f_X(x)\, dx \right) \log \left(\int_{R_i} f_X(x)\, dx \right);$$

if nothing else is know about the pdf $f_X(x)$, it's not possible to obtain any simpler exact expression for $H(I)$. However, we can make some progress and obtain some insight by focusing on uniform quantizers and adopting (as in Section 1.3) the high-resolution approximation

In the high-resolution regime of uniform quantizers (very large N, thus very small Δ), the probability of each cell $p_i = P[X \in R_i]$ can be approximated as

$$p_i \simeq f_X(y_i)\Delta,$$

because Δ is small enough to have $f_X(x)$ approximately constant inside R_i, and y_i is (approximately) the central point of R_i. In these conditions,

$$H(I) \simeq -\sum_{i=0}^{N-1} \Delta\, f_X(y_i) \log \left(\Delta f_X(y_i) \right),$$

with the approximation being more accurate as Δ becomes smaller. The expression above can be written as

$$H(I) \simeq \underbrace{-\sum_{i=0}^{N-1} \Delta\, f_X(y_i) \log \left(f_X(y_i) \right)}_{\simeq h(X)} - \log \Delta \underbrace{\sum_{i=0}^{N-1} \underbrace{\Delta\, f_x(y_i)}_{\simeq p_i}}_{\simeq 1} \simeq h(X) - \log \Delta,$$

where the first sum is approximately equal to $h(X)$ because as Δ approaches zero, the sum approaches the Riemmen integral of $f_X(x) \log f_X(x)$. In conclusion, the entropy of the output of a high-resolution uniform quantization encoder (in the high-resolution regime) is approximately equal to the differential entropy of the source, plus a term which depends on the precision (resolution) with which the samples of X are represented (quantized). Notice that as Δ becomes small, the term $-\log \Delta$ increases. If the output of the encoder is followed by an optimal entropic encoder (for example, using a Huffman code), the average number of bits, $\bar{L}$, used to encode each sample will be close to $H(I)$, that is $\bar{L} \simeq H(I)$.

The average rate $\bar{L}$ and the MSE are related through the pair of equalities

$$\bar{L} \simeq h(X) - \log \Delta \quad \text{and} \quad \text{MSE} = \frac{\Delta^2}{12};$$

these can be rewritten as

$$\bar{L} \simeq h(X) - \log \sqrt{12} - \frac{1}{2} \log \text{MSE} \quad \text{and} \quad \text{MSE} = \frac{1}{12}\, 2^{(2\,h(X) - 2\,\bar{L})},$$

showing that the average bit rate decreases logarithmically with the increase of the MSE and, conversely, the MSE decreases exponentially with the increase of the average bit rate.

Let us illustrate the results derived in the previous paragraph with a couple of simple examples. First, consider a random variable X with a uniform density on the interval $[a,\, b]$, that is $f_X(x) = 1/(b-a)$, if $x \in [a,\, b]$, and zero otherwise. The differential entropy of X is $h(X) = \log(b-a)$, thus,

$$\bar{L} \simeq H(I) \simeq \log(b-a) - \log \Delta = \log(b-a) - \log \frac{b-a}{N} = -\log(2^{-R}) = R \quad \text{bits/sample},$$

where all the logarithms are to base 2. The expression above means that, for a uniform density, the average number of bits per sample of a uniform high-resolution quantizer equals simply the quantizer rate R. This is regardless of the support of the density.

Now consider a triangular density on the interval $[0,\, 1]$, that is, $f_X(x) = 2 - 2x$, for $x \in [0,\, 1]$. In this case, it is easy to show that

$$h(X) = -\int_0^1 (2-2x) \log_2(2-2x)\, dx = \frac{1}{2} \log_2 \frac{e}{4} \simeq -0.279,$$

thus

$$\bar{L} \simeq -0.279 - \log \Delta = -0.279 - \log \frac{1}{N} = -0.279 - \log(2^{-R}) = R - 0.279 \quad \text{bits/sample}.$$

This example shows that if the density is not uniform on its support, then the average number of required bits per sample (after optimal entropic coding of the uniform high-resolution quantization encoder output) is less than the quantizer rate. This is a simple consequence of the fact that if the density is not uniform, then the cell probabilities $\{p_0, ..., p_{N-1}\}$ are not equal and the corresponding entropy is less than $\log N$. However, notice that we are in a high-resolution regime, thus $N \gg 1$ and the decrease in average bit rate caused by the non-uniformity of the density is relatively small.

2 Vector Quantization

2.1 Introduction and Definitions

In vector quantization the input to the encoder, that is, the output of the source to be quantized, is not a scalar quantity but a vector in $\mathbb{R}^n$. Formally, the source is modeled as a vector random

variable $\mathbf{X} \in \mathbb{R}^n$, characterized by a pdf $f_{\mathbf{X}}(\mathbf{x})$. Any pdf defined on $\mathbb{R}^n$ has to satisfy the following properties: $f_{\mathbf{X}}(\mathbf{x}) \geq 0$, for any $\mathbf{x} \in \mathbb{R}^n$,

$$\int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} = 1,$$

and

$$\int_R f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} = P[\mathbf{X} \in R],$$

where $P[\mathbf{X} \in R]$ denotes the probability that the random variable $\mathbf{X}$ takes values in some set $R \subseteq \mathbb{R}^n$. To avoid technical issues, we consider only continuous pdfs.

In the vector case, the encoder is a function $\mathcal{E} : \mathbb{R}^n \to \mathcal{I}$, where $\mathcal{I} = \{0, 1, ..., N-1\}$. As in the scalar case, this function is non-injective and there are many different values of the argument that produce the same value of the function; each of this sets is called a quantization region (or cell), and is defined as

$$R_i = \{\mathbf{x} \in \mathbb{R}^n : \mathcal{E}(\mathbf{x}) = i\}.$$

Since $\mathcal{E}$ is a function defined over all $\mathbb{R}^n$, this definition implies that the collection of quantization regions/cells $\mathcal{R} = \{R_0, ..., R_{N-1}\}$ defines a *partition* of $\mathbb{R}^n$, that is,

$$(i \neq j) \Rightarrow R_i \cap R_j = \emptyset \qquad \text{and} \qquad \bigcup_{i=0}^{N-1} R_i = \mathbb{R}^n. \tag{25}$$

The decoder is a function $\mathcal{D} : \mathcal{I} \to \mathbb{R}^n$; as in the scalar case, since the argument of $\mathcal{D}$ only takes N different values, and $\mathcal{D}$ is a deterministic function, it can also only take N different values, thus its range is a finite set $\mathcal{C} = \{\mathbf{y}_0, ..., \mathbf{y}_{N-1}\} \subset \mathbb{R}^n$. The set $\mathcal{C}$ is still called the codebook. The i-th element of the codebook, $\mathbf{y}_i$, is the *representative* of the region/cell R_i.

Considering that there are no errors in the channel, the sample $\mathbf{x}$ is reproduced by the decoder as $\mathcal{D}(\mathcal{E}(\mathbf{x}))$, that is, the result of first encoding and then decoding x. The composition of the functions $\mathcal{E}$ and $\mathcal{D}$ defines the so-called *vector quantization function* $\mathcal{Q} : \mathbb{R}^n \to \mathcal{C}$, where $\mathcal{Q}(\mathbf{x}) = \mathcal{D}(\mathcal{E}(\mathbf{x}))$. As in the scalar case, the quantization function has the following obvious property

$$(\mathbf{x} \in R_i) \Leftrightarrow \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i. \tag{26}$$

Similarly to the scalar case, a vector quantizer (VQ) (equivalently a pair encoder/decoder) is completely defined by the set of regions $\mathcal{R} = \{R_0, ..., R_N\}$ and the corresponding codebook $\mathcal{C} = \{\mathbf{y}_0, ...\mathbf{y}_{N-1}\} \subset \mathbb{R}^n$. A VQ in which all the cells are convex and contain its representative is called a regular VQ. Recall that a set S is said to be convex if it satisfies the condition

$$\mathbf{x}, \mathbf{y} \in S \;\; \Rightarrow \;\; \lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in S, \quad \text{for any } \lambda \in [0, 1];$$

in words, a set is convex when the line segment joining any two of its points also belongs to the set. Observe that this definition covers the scalar case, since the only type of convex sets in $\mathbb{R}$ are intervals (regardless of being open or close). Figure 1 illustrates the concepts of convex and non-convex sets.

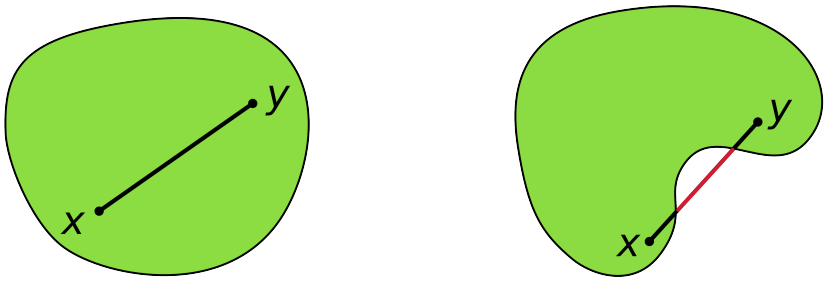

Figure 1: A convex set (left) and a non-convex set (right).

2.2 Optimal VQs, Lloyd's Algorithm, and the Linde-Buzo-Gray Algorithm

This subsection is parallel to Section 1.2, essentially repeating all the concepts and derivations, adapted to the vectorial case.

2.2.1 Expected Distortion and the Optimal VQ

Finding an optimal VQ consists in finding the set of regions, $\mathcal{R}$, and the codebook, $\mathcal{C}$, that minimizes a given objective function. Although there are other options, the standard choice is the MSE

$$\text{MSE} = E\left[\|\mathbf{X} - \mathcal{Q}(\mathbf{X})\|^2\right] = \int_{\mathbb{R}^n} f_X(\mathbf{x}) \|\mathbf{x} - \mathcal{Q}(\mathbf{x})\|^2 \, d\mathbf{x},$$

where $\|\mathbf{v}\|^2 = \sum_i^n v_i^2$ denotes the usual squared Euclidean norm of some vector $\mathbf{v} \in \mathbb{R}^n$. Some authors define the MSE of a VQ in $\mathbb{R}^n$ with a $1/n$ factor, that is, $\text{MSE} = (1/n)E\left[\|\mathbf{X} - \mathcal{Q}(\mathbf{X})\|_2^2\right]$, which in this case becomes a measure of average quadratic error *per coordinate*. In this text, we will not adopt that convention.

Adopting the MSE to measure the quantizer performance, the problem of finding the optimal set of regions and corresponding representatives becomes

$$\left(\mathcal{R}^{\text{opt}}, \mathcal{C}^{\text{opt}}\right) = \arg\min_{\mathcal{R},\mathcal{C}} \sum_{i=0}^{N-1} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \|\mathbf{x} - \mathbf{y}_i\|^2 \, d\mathbf{x}, \tag{27}$$

which is similar to (3)-(4), but here for the vectorial case.

2.2.2 Partial Solutions

As in the scalar case, it is possible to solve the two partial problems:

- Given the quantization regions regions $\mathcal{R} = \{R_0, ..., R_{N-1}\}$, find the corresponding optimal codebook,

$$\{\mathbf{y}_0^*, ..., \mathbf{y}_{N-1}^*\} = \arg\min_{\mathbf{y}_0^*, ..., \mathbf{y}_{N-1}} \sum_{i=0}^{N-1} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \|\mathbf{x} - \mathbf{y}_i\|^2 \, d\mathbf{x}. \tag{28}$$

- Given a codebook $\mathcal{C} = \{\mathbf{y}_0, ..., \mathbf{y}_{N-1}\}$, find the optimal regions,

$$\{R_0^*, ..., R_{N-1}^*\} = \arg \min_{R_0^*,...,R_{N-1}} \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \, \|\mathbf{x} - \mathcal{Q}(\mathbf{x})\|^2 \, d\mathbf{x} \tag{29}$$

$$\text{subject to} \quad (i \neq j) \Rightarrow R_i \cap R_j \tag{30}$$

$$\bigcup_{i=0}^{N-1} R_i = \mathbb{R}. \tag{31}$$

In (28), the function being minimized is the sum of N non-negative functions, each one of them only dependent on one of the $\mathbf{y}_i$. The problem can be decoupled into N independent problems

$$\mathbf{y}_i^* = \arg \min_{\mathbf{y}} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \|\mathbf{x} - \mathbf{y}\|_2^2 \, d\mathbf{x}.$$

Expanding the squared Euclidean norm into $\|\mathbf{x} - \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle$, leads to

$$\mathbf{y}_i^* = \arg \min_{\mathbf{y}} \left[\int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \|\mathbf{x}\|_2^2 \, d\mathbf{x} + \|\mathbf{y}\|_2^2 \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x} - 2 \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \langle \mathbf{y}, \mathbf{x} \rangle \, d\mathbf{x} \right] \tag{32}$$

$$= \arg \min_{\mathbf{y}} \left[\|\mathbf{y}\|_2^2 \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x} - 2 \left\langle \mathbf{y}, \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \mathbf{x} \, d\mathbf{x} \right\rangle \right], \tag{33}$$

where the second equality is due to the fact that the first term in (32) does not depend on $\mathbf{y}$, thus it is irrelevant for the minimization, and the inner product commutes with integration (since both are linear operators). The minimum is found by computing the gradient with respect to $\mathbf{y}$ and equating to zero. Recalling that $\nabla_{\mathbf{v}} \|\mathbf{v}\|^2 = 2\mathbf{v}$ and $\nabla_{\mathbf{v}} \langle \mathbf{v}, \mathbf{b} \rangle = \mathbf{b}$, we have

$$\nabla_{\mathbf{y}} \left[\|\mathbf{y}\|_2^2 \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x} - 2 \, \mathbf{y} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \mathbf{x} \, d\mathbf{x} \right] = 2 \, \mathbf{y} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) - 2 \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \mathbf{x} \, d\mathbf{x}$$

Equating to zero, leads to the following equation

$$\mathbf{y} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) = \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \mathbf{x} \, d\mathbf{x},$$

the solution of which is

$$\mathbf{y}_i^* = \frac{\displaystyle\int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, \mathbf{x} \, d\mathbf{x}}{\displaystyle\int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x}}. \tag{34}$$

As in the scalar case, (34) has a clear probabilistic meaning: it is the conditional expected value of the random variable $\mathbf{X}$, given that $\mathbf{X}$ is in R_i. A more *physical* interpretation of (34) is that it is the center of (probabilistic) mass of region R_i.

The partial problem (29) has similar solution to (6): given a codebook $\mathcal{C} = \{\mathbf{y}_0, ..., \mathbf{y}_{N-1}\}$, the best possible encoder is one that chooses, for each $\mathbf{x}$, the closest representative. In conclusion, the optimal regions are given by

$$R_i = \{\mathbf{x} : \, \|\mathbf{x} - \mathbf{y}_i\|^2 \leq \|\mathbf{x} - \mathbf{y}_j\|^2, \, j \neq i\}, \quad \text{for} \quad i = 0, ..., N-1, \tag{35}$$

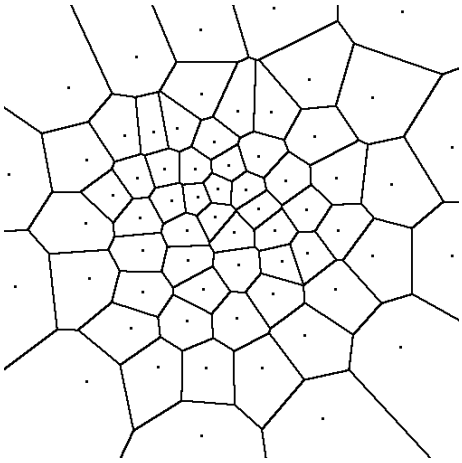

Figure 2: Example of Voronoi regions for a set of points in $\mathbb{R}^2$.

that is, R_i is the set of points that are closer to $\mathbf{y}_i$ than to any other element of the codebook. Whereas in the scalar case these regions where simply intervals, in $\mathbb{R}^n$ the optimal regions may have a more complex structure. The N regions that partition $\mathbb{R}^n$ according to (35) are called the Voronoi regions (or Dirichlet tessellation) corresponding to the set of points $\{\mathbf{y}_0, ..., \mathbf{y}_{N-1}\}$. An important property of Voronoi regions (the proof is beyond the scope of this text) is that they are necessarily convex, thus a Lloyd vector quantizer is necessarily regular. Figure 2 illustrates the concept of Voronoi regions in $\mathbb{R}^2$.

2.2.3 The Lloyd Algorithm

The Lloyd algorithm for VQ design works exactly as the scalar counterpart.

Step 1: Given the current codebook $\mathcal{C}^{(t)} = \left\{\mathbf{y}_0^{(t)}, ..., \mathbf{y}_{N-1}^{(t)}\right\}$, obtain the optimal regions

$$R_i^{(t)} = \{\mathbf{x} : \|\mathbf{x} - \mathbf{y}_i^{(t)}\|^2 \leq \|\mathbf{x} - \mathbf{y}_j^{(t)}\|^2, \, j \neq i\}, \quad \text{for } \; i = 0, ..., N-1;$$

Step 2: Given the current regions $\mathcal{R}^{(t)} = \left\{R_0^{(t)}, ..., R_{N-1}^{(t)}\right\}$, update the representatives

$$\mathbf{y}_i^{(t+1)} = \frac{\displaystyle\int_{R_i^{(t)}} f_{\mathbf{X}}(\mathbf{x})\, \mathbf{x}\, d\mathbf{x}}{\displaystyle\int_{R_i^{(t)}} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}}, \quad \text{for } \; i = 0, ..., N-1;$$

Step 3: Check some stopping criterion; if it is satisfied, stop; if not, set $t \leftarrow t+1$, and go back to Step 1.

A typical stopping criterion would be to check if the maximum squared distance between two consecutive positions of codebook elements is less than some threshold; that is, the algorithm would be stopped if the following condition is satisfied

$$\max_i \|\mathbf{y}_i^{(t)} - \mathbf{y}_i^{(t+1)}\|^2 \leq \varepsilon. \tag{36}$$

2.2.4 Zero Mean Quantization Error of Lloyd Quantizers

The property of scalar Lloyd quantizers shown in Subsection 1.2.5 (that the quantization error has zero mean) is still valid in the vectorial case. Notice that the derivation carried out in Subsection 1.2.5 can be directly applied in $\mathbb{R}^n$. Thus, it is still true that for a VQ that satisfies the conditions (34) and (35), called Lloyd VQs, the mean of the quantization error is zero, that is, $E\left[\mathcal{Q}(\mathbf{X}) - \mathbf{X}\right] = 0$, or, equivalently, $E[\mathcal{Q}(\mathbf{X})] = E[\mathbf{X}]$.

2.2.5 The Linde-Buzo-Gray Algorithm

The Linde-Buzo-Gray algorithm for the vector case has exactly the same structure as in the scalar case, so it will not be described again. The only practical detail which is significantly different in the vectorial case is an increased sensitivity to initialization; thus, when using this algorithm to obtain a VQ, care has to be taken in choosing the initialization of the algorithm. For further details on this and other aspects of the LBG algorithm, the interested reader is referred to [2].

2.3 High-Resolution Approximation

2.3.1 General Case

As in the scalar case, it is possible to obtain approximate expressions for the MSE of high-resolution VQs, from which some insight into their performance may be obtained. In the high-resolution regime, just as in the scalar case, the key assumption is that the regions/cells are small enough to allow approximating the pdf of $\mathbf{X}$ by a constant inside each region. With this approximation, the MSE can be written as

$$\begin{aligned} \text{MSE} &= \sum_{i=0}^{N-1} \int_{R_i} f_{\mathbf{X}}(\mathbf{x}) \left\|\mathbf{x} - \mathbf{y}_i\right\|^2 d\mathbf{x} \\ &\simeq \sum_{i=0}^{N-1} f_{\mathbf{X}}(\mathbf{y}_i) \int_{R_i} \left\|\mathbf{x} - \mathbf{y}_i\right\|^2 d\mathbf{x} \\ &= \sum_{i=0}^{N-1} f_{\mathbf{X}}(\mathbf{y}_i) V_i \int_{R_i} \frac{1}{V_i} \left\|\mathbf{x} - \mathbf{y}_i\right\|^2 d\mathbf{x}. \end{aligned} \tag{37}$$

where $V_i = V(R_i) \equiv \int_{R_i} d\mathbf{x}$ is the volume (area, in $\mathbb{R}^2$, length in $\mathbb{R}$) of region R_i. Noticing that $p_i = P[\mathbf{X} \in R_i] \simeq f_{\mathbf{X}}(\mathbf{y}_i) V_i$, we have

$$\text{MSE} \simeq \sum_{i=0}^{N-1} p_i \, \frac{\displaystyle\int_{R_i} \left\|\mathbf{x} - \mathbf{y}_i\right\|^2 d\mathbf{x}}{\displaystyle\int_{R_i} d\mathbf{x}} = \sum_{i=0}^{N-1} p_i \, \frac{1}{V_i} \int_{R_i} \left\|\mathbf{x} - \mathbf{y}_i\right\|^2 d\mathbf{x}. \tag{38}$$

Unlike in the scalar case, where the quantity multiplying each p_i can be shown to be $\Delta_i^2/12$, the involved integration not always has closed form expressions, or can even be computed exactly.

However, if we are in the presence of a Lloyd quantizer, $\mathbf{y}_i$ is the center of mass of region R_i, thus the quantity

$$\frac{1}{V_i} \int_{R_i} \|\mathbf{x} - \mathbf{y}_i\|^2 \, d\mathbf{x} \tag{39}$$

can be recognized as the the moment of inertia of the region R_i about its center of mass, if the total mass is one and the density is uniform.

2.3.2 Uniform VQ

To make some progress, we now assume that we are in the presence of a uniform VQ, that is, such that all the regions have a similar shape and size; in other words, the regions R_0, $R_1, ..., R_{N-1}$ only differ from each other by a shift of location. In this condition, it clear that the value of both the numerator and the denominator of (39) is the same for all cells: the denominator is simply the volume, which of course does not depend on the location; the numerator, after the change of variable $\mathbf{z} = \mathbf{x} - \mathbf{y}_i$, can be written, for any i, as

$$\int_{R_i} \|\mathbf{x} - \mathbf{y}_i\|^2 \, d\mathbf{x} = \int_R \|\mathbf{z}\|^2 \, d\mathbf{z},$$

where R denotes a region with the same volume and shape as all the R_i's, but such that the center of mass is at the origin. The MSE expression thus simplifies to

$$\text{MSE} \simeq \frac{1}{V(R)} \int_R \|\mathbf{x}\|^2 \, d\mathbf{x} \sum_{i=1}^{N} p_i = \frac{1}{V(R)} \int_R \|\mathbf{x}\|^2 \, d\mathbf{x}. \tag{40}$$

The expression (40) shows that the MSE of a high-resolution uniform VQ depends only on the volume and the shape of the quantization cells. This can be made even more explicit by re-writing it as

$$\text{MSE} \simeq V(R)^{2/n} \underbrace{\left(\left(\frac{1}{V(R)} \right)^{2/n} \frac{\int_R \|\mathbf{x}\|^2 \, d\mathbf{x}}{V(R)} \right)}_{\text{depends only on the shape}} = V(R)^{2/n} \, M(R), \tag{41}$$

where the second factor, denoted $M(R)$, depends only on the shape (not the volume, as we will prove next) and the first factor, $V(R)^{2/n}$, depends only on the volume. To prove that $M(R)$, called the normalized moment of inertia, is independent of the volume, we show that it is invariant to a change of scale, that is, $M(cR) = M(R)$, for any $c \in \mathbb{R}_+$:

$$M(cR) = \left(\frac{1}{V(cR)} \right)^{2/n} \frac{1}{V(cR)} \int_{cR} \|\mathbf{x}\|^2 \, d\mathbf{x} \tag{42}$$

$$= \left(\frac{1}{c^n \, V(R)} \right)^{2/n} \frac{1}{c^n \, V(R)} \int_R \|c\mathbf{z}\|^2 \, c^n \, d\mathbf{z} \tag{43}$$

$$= \quad c^{-2}\, c^{-n}\, c^{2+n} \underbrace{\left(\frac{1}{V(R)}\right)^{2/n} \frac{1}{V(R)} \int_R \|\mathbf{z}\|^2 \, d\mathbf{z}}_{M(R)} \tag{44}$$

$$= \quad M(R). \tag{45}$$

The volume of the regions (which is the same for all regions in a uniform quantizer) depends only on the number of regions and on the volume of the support of $f_{\mathbf{X}}(\mathbf{x})$, denoted $V(B)$. Of course, for a source $\mathbf{X}$ with unbounded support (for example, a Gaussian), the support is the whole space $B = \mathbb{R}^n$, and this reasoning does not apply exactly. However, as in the scalar case, we can identify some region outside of which the probability of finding $\mathbf{X}$ is arbitrarily small, and consider that as the support B. For a given support, the region volume $V(R)$ will be simply the total volume of the support, divided by the number of regions, that is,

$$V(R) = \frac{V(B)}{N} = V(B)\, 2^{-R}. \tag{46}$$

Inserting this expression in (41) leads to

$$MSE \simeq V(B)^{2/n}\, 2^{-2R/n}\, M(R), \tag{47}$$

showing that, as in the scalar case (see (23)), the MSE also decreases exponentially with R. In the scalar case, we have $n = 1$ and (46) becomes similar to (23)

$$MSE \simeq V(B)^2\, 2^{-2R}\, M(R),$$

where we identify the volume of the support as $V(B) = A$ and $M(R) = 1/12$.

However, for $n > 1$, the MSE decreases slower as R increases, since the exponent is $-2R/n$; for example, in $\mathbb{R}^2$, each extra bit only decreases the MSE by a factor of 2 (instead of 4 in the scalar case); as another example, in $\mathbb{R}^{20}$, we have $2^{1/10} \simeq 1.0718$, thus each extra bit only reduces the MSE by a factor of approximately 1.0718. In logarithmic units, we can write (as in Section 1.3.1)

$$SNR \simeq (K + (6/n)R)\, dB$$

showing that each extra bit in the quantizer resolution, achieves an improvement of approximately $(6/n)dB$ in the quantization SNR.

Concerning the "shape factor" $M(R)$, there is a crucial difference between the scalar case ($n = 1$) and the vector case ($n > 1$). In the scalar case, the only possible convex set is an interval, and it's easy to verify that $M(R) = 1/12$. However, for $n > 1$, we have some freedom in choosing the shape of the quantization cells, under the constraint that this shape allows a partition (or tesselation) of the support S.

2.3.3 Optimal Tesselations

After decoupling the high-resolution approximation of the MSE into a factor that depends only on the volume of the regions (thus on the number of regions) and another factor that depends only on the shape, we can concentrate on studying the effect of the region shapes.

It is known that the shape with the smallest moment of inertia, for a given mass and volume, is a sphere (a circle, in $\mathbb{R}^2$). Although spherical regions can not be used, because they do not partition the space, they provide a lower bound on the moment of inertia. Let us thus compute the factor $M(R)$, when R is a sphere (a circle) in $\mathbb{R}^2$; since, as seen above, this quantity does not depend on the size of the region, we consider unit radius.

The volume of a sphere of unit radius in $\mathbb{R}^n$, denoted S_n, is known to be

$$V(S_n) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)},$$

where Γ denote's Euler's gamma function. For $n = 2$, since $\Gamma(2) = 1$, we obtain $V(S_2) = \pi$, which is the well known area of a unit circle. For $n = 3$, since $\Gamma(3/2) = 3\sqrt{\pi}/4$, we obtain the also well-known volume of a 4-dimensional sphere, $V(S_3) = 4\pi/3$.

The other quantity needed to obtain $M(S_2)$ (see (44)) is

$$\int_{S_2} \|\mathbf{z}\|^2 \, d\mathbf{z},$$

which is more convenient to compute in polar coordinates, that is,

$$\begin{aligned} \int_{C_2} \|\mathbf{z}\|^2 \, d\mathbf{z} &= \int_0^{2\pi} \int_0^1 \rho^2 \, \rho \, d\rho \, d\theta && (48) \\ &= 2\,\pi \int_0^1 \rho^3 \, d\rho && (49) \\ &= \frac{\pi}{2}. && (50) \end{aligned}$$

Plugging these results into the definition of $M(S_2)$ (see (44)), we finally obtain

$$M(S_2) = \frac{1}{V(S_2)^2} \int_{S_2} \|\mathbf{z}\|^2 \, d\mathbf{z} = \frac{1}{2\pi} \simeq 0.159155. \quad (51)$$

Let us now compute $M(C_2)$, where C_n denotes the cubic region of unit side in $\mathbb{R}^n$; for $n = 2$, this is a square of unit side. Of course, $V(C_n) = 1$, for any n, since the volume of a cube of side d in $\mathbb{R}^n$ is simply d^n. As for the quantity

$$\int_{C_2} \|\mathbf{z}\|^2 \, d\mathbf{z},$$

the integration can be carried out easily as follows:

$$\begin{aligned} \int_{C_2} \|\mathbf{z}\|^2 \, d\mathbf{z} &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} z_1^2 + z_2^2 \, dz_1 \, dz_2 && (52) \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} z_1^2 \, dz_1 \, dz_2 + \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} z_2^2 \, dz_1 \, dz_2 && (53) \\ &= 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} z_1^2 \, dz_1 && (54) \\ &= \frac{1}{6}. && (55) \end{aligned}$$

Since $V(C_2) = 1$, we have $M(C_2) = 1/6 \simeq 0.166667$, showing that, as expected, using square quantization regions leads to a higher quantization noise than what would be obtained if circular regions could be used (which they can not).

The fundamental questions are: is there any other shape with which it is possible to cover $\mathbb{R}^n$ and which leads to a smaller MSE (that is, has a lower moment of inertia)? is there an optimal shape? In $\mathbb{R}^2$, the answer to these questions is positive: yes, the optimal shape is a regular hexagon. For general $\mathbb{R}^n$, with $n > 2$, the answer to these questions is still an open problem. The proof of optimality of the hexagonal VQ is beyond the scope of this text; however, we can compute $M(H)$, where H denotes an hexagonal region centered at the origin, and confirm that it is lower than $M(C_2)$ but larger than $M(S_2)$.

Since $M(H)$ does not depend on the size of H, we consider an hexagon with unit apothem, $h = 1$ (recall that the apothem is the distance from the center to the mid-point of one of the sides). In this case, using the well-known formula for the area of a regular polygon as a function of the apothem,

$$V(H) = h^2\, 6 \tan\left(\frac{\pi}{6}\right) = \frac{6}{\sqrt{3}}.$$

Finally, to compute the integral of $\|\mathbf{z}\|^2$ over the hexagon, we notice that this function has circular symmetry and that the hexagon can be split into 12 similar triangles, one of which is given by $T = \{\mathbf{z} = (z_1,\, z_2) :\, 0 \le z_1 \le 1 \text{ and } 0 \le z_2 \le z_1/\sqrt{3}\}$. Consequently,

$$\begin{aligned}
\int_H \|\mathbf{z}\|^2\, d\mathbf{z} &= 12 \int_0^1 \int_0^{z_1/\sqrt{3}} z_1^2 + z_2^2\, dz_2\, dz_1 && (56)\\
&= 12 \int_0^1 z_1^2 \underbrace{\int_0^{z_1/\sqrt{3}} dz_2}_{z_1/\sqrt{3}}\, dz_1 + 12 \int_0^1 \underbrace{\int_0^{z_1/\sqrt{3}} z_2^2\, dz_2}_{z_1^3/(9\sqrt{3})}\, dz_1 && (57)\\
&= \frac{12}{\sqrt{3}} \underbrace{\int_0^1 z_1^3\, dz_1}_{=1/4} + \frac{12}{9\sqrt{3}} \underbrace{\int_0^1 z_1^3\, dz_1}_{=1/4} && (58)\\
&= \frac{10}{3\sqrt{3}} && (59)
\end{aligned}$$

Combining this quantity with the volume $V(H) = 6/\sqrt{3}$, we finally have

$$M(H) = \frac{1}{V(H)^2} \int_H \|\mathbf{z}\|^2\, d\mathbf{z} = \left(\frac{\sqrt{3}}{6}\right)^2 \frac{10}{3\sqrt{3}} = \frac{5}{18\sqrt{3}} \simeq 0.160375.$$

Comparing this value with the previous ones ($M(S_2) \simeq 0.159155$ and $M(C_2) \simeq 0.166667$), we can conclude that the hexagonal VQ is indeed better than the cubical one, with a normalized moment of inertia only 0.7% larger than that of a circle (which can't be used, as explained above).

References

[1] Q. Du, M. Emelianenko, and L. Ju, "Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations", *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 102–119, 2006.

[2] A. Gersho and R. Gray, "Vector Quantization and Signal Compression." *Kluwer Acadmic Publishers*, 1992.

[3] R. Gray, "Vector quantization", *Acoustics, Speech, and Signal Processing Magazine,* vol. 1, no. 2, 1984.

[4] R. Gray and D. Neuhoff, "Quantization", *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.