

1 System Design and Implementation

In the course of this project, we began by implementing a C program to perform simple Vector Quantization, and adapted it to an implementation of Channel-Optimized Vector Quantization by modifying the optimality conditions as was outlined in the Description section. We then wrote a C implementation of the I→J system without a channel. To interface with images, we wrote a Python wrapper program. We also used GNU Octave, MathWorks MATLAB, and Python to produce data and create plots.

In the subsequent sections we will detail the design challenges that were faced along the way and how they were addressed.

1.1 Conditions of Optimality

Expanding for the optimality expressions given in [Description], we get that the optimal reconstruction of X , given fixed encoders and received codebook indices k, l is given by

$$x_{(k,l)} = \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} E[X|X \in R_i, Y \in S_j] P(k, l|i, j)}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(X \in R_i, Y \in S_j) P(k, l|i, j)} \quad (1)$$

$$= \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(k, l|i, j) \int_{x \in R_i} x \frac{f_X(x)}{P(X \in R_i, Y \in S_j)} dx}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(X \in R_i, Y \in S_j) P(k, l|i, j)} \quad (2)$$

Let \mathcal{T} be a finite training set of ordered pairs drawn from the joint distribution for X and Y . Letting M_{ij} be the number of training vectors in $R_i \times S_j$ we can thus approximate the integral and probabilities as follows:

$$x_{(k,l)} = \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(k, l|i, j) \sum_{x \in R_i} x \frac{1}{M_{ij}}}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(X \in R_i, Y \in S_j) P(k, l|i, j)} \quad (3)$$

By symmetry, the optimal reconstruction for Y is given by

$$y_{(k,l)} = \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(k, l|i, j) \sum_{y \in S_j} y \frac{1}{M_{ij}}}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(X \in R_i, Y \in S_j) P(k, l|i, j)} \quad (4)$$

Similarly, we write out the nearest neighbour conditions; the idea is the same as in COVQ. The optimal encoding partitions for X given fixed partitions in Y , and fixed decoding points $\{(x_{(k,l)}, y_{(k,l)})\}$ are given by:

$$R_i = \{x | d(x, i) \leq d(x, h), \forall h \in \{1, \dots, N_X\}\} \quad (5)$$

and symmetrically, the optimal encoding partitions for Y given fixed partitions in X , and fixed decoding points $\{(x_{(k,l)}, y_{(k,l)})\}$ are given by::

$$S_j = \{y | d(y, j) \leq d(y, h), \forall h \in \{1, \dots, N_Y\}\} \quad (6)$$

where $d(x, i)$ is given by

$$d(x, i) = E[Y^2|X = x] + \sum_{j=1}^{N_Y} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} ((x - x_{(k,l)})^2 - 2y_{(k,l)}E[Y|X = x, Y \in S_j] + y_{(k,l)}^2)P(Y \in S_j|X = x)P(k, l|i, j) \quad (7)$$

and $d(y, j)$ is given by

$$d(y, j) = E[X^2|Y = y] + \sum_{i=1}^{N_X} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} ((y - y_{(k,l)})^2 - 2x_{(k,l)}E[X|Y = y, X \in R_i] + x_{(k,l)}^2)P(X \in R_i|Y = y)P(k, l|i, j) \quad (8)$$

Here, we note how the nearest neighbour condition for X depends on the Y encoding regions $\{S_j\}$, and vice versa. We therefore adapt the Lloyd iteration into three stages. This poses a design issue with respect to (a) order of iteration and (b) initialization of codebook. The former we will address at the end of this section, the latter in the next section.

Focusing on $d(x, i)$, since $d(y, j)$ is analogous, we write out the expectations:

$$d(x, i) = \int_{-\infty}^{\infty} y^2 f_{Y|X}(y|X = x) dy + \sum_{j=1}^{N_Y} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} ((x - x_{(k,l)})^2 - 2y_{(k,l)} \int_{-\infty}^{\infty} y^2 \frac{f_{Y|X}(y|X = x)}{P(Y \in S_j)} dy + y_{(k,l)}^2) \int_{S_j} y f_{Y|X}(y|X = x) dy \cdot P(k, l|i, j) \quad (9)$$

The part of this expression that poses a new problem is $f_{Y|X}(y|X = x)$. In order to approximate this with a finite training set, we can first quantize the training values with two one-dimensional uniform quantizers. This is formalized below.

Define $q_X(x) : \mathbb{R} \rightarrow \{x_1, \dots, x_{L_X}\}$ and $q_Y(y) : \mathbb{R} \rightarrow \{y_1, \dots, y_{L_Y}\}$ to be the L_X - and L_Y -level uniform quantizers for the sources X and Y respectively, where the $x_i - x_{i-1}$ is constant for all $i \in \{2, \dots, L_X\}$, and $y_i - y_{i-1}$ constant for all $i \in \{2, \dots, L_Y\}$

$$q_X(x) = x_i \in \{x_1, \dots, x_{L_X}\} \iff x \in \left[\frac{x_{i-1} + x_i}{2}, \frac{x_i + x_{i+1}}{2} \right) \quad (10)$$

$$q_Y(y) = y_i \in \{y_1, \dots, y_{L_Y}\} \iff y \in \left[\frac{y_{i-1} + y_i}{2}, \frac{y_i + y_{i+1}}{2} \right) \quad (11)$$

with the convention that $x_0 = y_0 = -\infty$, and $x_{L_X+1} = y_{L_Y+1} = +\infty$.

For simplicity of design, we attach this uniform quantizer to the beginning of our system. That is, we essentially transform the sources to have state spaces of size L_X, L_Y before performing the I→J quantization. (Modified block diagram)

We can now explicitly rewrite our approximated conditions in computable terms via a uniformly quantized training set. First, we introduce some terms to simplify notation. Let $m(x_g, y_h)$ be the number of training pairs that quantize to the pair (x_g, y_h) :

$$m(x_g, y_h) = |\{(x, y) \in \mathcal{T} : q_X(x) = x_g, q_Y(y) = y_h\}| \quad (12)$$

$$\forall g \in \{1, \dots, L_X\}, \forall h \in \{1, \dots, L_Y\}$$

Let $I_X(x_g) \in \{1, \dots, N_X\}$, $I_Y(y_h)$ be the index of ‘closest’ X codevector, $I_Y(y_h) \in \{1, \dots, N_Y\}$ be the index of ‘closest’ Y codevector:

$$I_X(x_g) = \arg \min_{i \in \{1, \dots, N_X\}} d_X(x_g, i) \quad (13)$$

$$I_Y(y_h) = \arg \min_{j \in \{1, \dots, N_Y\}} d_Y(y_h, j) \quad (14)$$

Notice that these are simply the nearest neighbour conditions for the uniformly quantized training set.

Let $M(x_g), M(y_h)$ be the number of training pairs that uniformly quantize to pairs of the form $(x_g, *)$, $(*, y_h)$ respectively:

$$M(x_g) = \sum_{h=1}^{L_Y} m(x_g, y_h) \quad (15)$$

$$M(y_h) = \sum_{g=1}^{L_X} m(x_g, y_h) \quad (16)$$

let $M_j(x_g)$ be the number of training pairs that uniformly quantize to pairs of the form (x_g, y_h) with y_h in the encoding region S_j , and $M_i(y_h)$ be the number of training pairs that uniformly quantize to pairs of the form (x_g, y_h) with x_g in the encoding region R_i :

$$M_j(x_g) = \sum_{y_h: I_Y(y_h)=j} m(x_g, y_h) \quad (17)$$

$$M_i(y_h) = \sum_{x_g: I_X(x_g)=i} m(x_g, y_h) \quad (18)$$

let $S_j(x_g), S_i(y_h)$ be given by:

$$S_j(x_g) = \sum_{y_h: I_Y(y_h)=j} y_h \cdot m(x_g, y_h) \quad (19)$$

$$S_i(y_h) = \sum_{x_g: I_X(x_g)=i} x_g \cdot m(x_g, y_h) \quad (20)$$

$$(21)$$

and $T(x_g), T(y_h)$ be given by:

$$T(x_g) = \sum_{y_h=1}^{L_Y} y_h^2 \cdot m(x_g, y_h) \quad (22)$$

$$T(y_h) = \sum_{x_g=1}^{L_X} x_g^2 \cdot m(x_g, y_h) \quad (23)$$

We now rewrite d_X, d_Y in the above terms. For shorthand, let $q_X(x) = x_g \in \{x_1, \dots, x_{L_X}\}$ and $q_Y(y) = y_h \in \{y_1, \dots, y_{L_Y}\}$

$$d_X(x, i) = \frac{1}{M(x_g)} \left(T(x_g) + \sum_{j=1}^{N_Y} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} \left(((x - x_{(k,l)})^2 + y_{(k,l)}^2) M_j(x_g) - 2y_{(k,l)} S_j(x_g) \right) P(k, l|i, j) \right) \quad (24)$$

$$d_Y(y, j) = \frac{1}{M(y_h)} \left(T(y_h) + \sum_{i=1}^{N_X} \sum_{k=1}^{N_X} \sum_{l=1}^{N_Y} \left(((y - y_{(k,l)})^2 + x_{(k,l)}^2) M_i(y_h) - 2x_{(k,l)} S_j(y_h) \right) P(k, l|i, j) \right) \quad (25)$$

(24) and (25) are now easily computable for given pair (x, i) or (y, j) , so the two nearest neighbour optimization steps reduce to performing a search over codebook indices i, j respectively, for the solution to equations (13), (14) respectively.

We also note that, besides aiding notation, the terms $M_j(x_g), S_j(x_g), T(x_g)$, and their counterparts $M_i(y_h), S_i(y_h), T(y_h)$ can be computed initially, independent of x_g, y_h respectively, reducing the computational complexity of the nearest neighbour lookup.

We now write out the centroid condition in a similar manner. We reiterate that the encoding partitions $\{R_i\}_{i=1}^{N_X}, \{S_j\}_{j=1}^{N_Y}$ are held constant in the application of this condition.

Let $M_{(i,j)}$ be the number of training vectors that belong to $R_i \times S_j$:

$$M_{(i,j)} = \sum_{\substack{x_g: I_X(x_g)=i \\ y_h: I_Y(y_h)=j}} m(x_g, y_h) \quad (26)$$

and let $S_{(i,j)}^X, S_{(i,j)}^Y$ be given by:

$$S_{(i,j)}^X = \sum_{\substack{x_g: I_X(x_g)=i \\ y_h: I_Y(y_h)=j}} x_g \cdot m(x_g, y_h) \quad (27)$$

$$S_{(i,j)}^Y = \sum_{\substack{x': I_X(x')=i \\ y': I_Y(y')=j}} y_h \cdot m(x_g, y_h) \quad (28)$$

With the above notation, the reconstructions that minimize end-to-end distortion are:

$$x_{(k,l)} = \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} S_{(i,j)}^X P(k, l|i, j)}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} M_{(i,j)} P(k, l|i, j)} \quad (29)$$

$$y_{(k,l)} = \frac{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} S_{(i,j)}^Y P(k, l|i, j)}{\sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} M_{(i,j)} P(k, l|i, j)} \quad (30)$$

Equations (29), (30) are now easily computable when our encoders are fixed.

Adding the uniform quantizer to the system presents new parameters. These are namely the choice of $\{x_1, y_1, x_{L_X}, y_{L_Y}, L_X, L_Y\}$. Together, these parameters control the granularity of quantization, and position

of the quantizer. For our simulations, we chose to (a) center the quantizer about the sample mean of T , (b) pick $\{x_1, y_1, x_{L_X}, y_{L_Y}\}$ such that all of T lies within the square defined by the four terms. With these decisions, the added complexity is simply how fine the quantizer ought to be.

For a given training set T , we want fine enough quantization so to not distort the equations (24) and (25), but coarse enough so that the conditional probabilities are well-defined for most bins. It is clearly the case that with a larger training set T , we can use finer quantizations while maintaining well-defined bins. For our purposes, we iterate through values of L_X, L_Y , choosing the result with lowest average distortion.

It is also important to note that the computational complexity depends primarily on the numbers L_X, L_Y , not the size of T , as was the case in the I→I system.

As was mentioned in this section, the fact that there are now three necessary conditions for optimality necessitates a decision in the order of their application in the quantizer training stage. We chose to give equal opportunities to each condition, beginning with one of the encoders. There are clearly other possible rotation schemes that might provide an advantage. These are not explored in this paper.

1.2 Codebook Initialization

1.3 Codeword Assignment

1.4 Image Coding

1.5 Design Summary