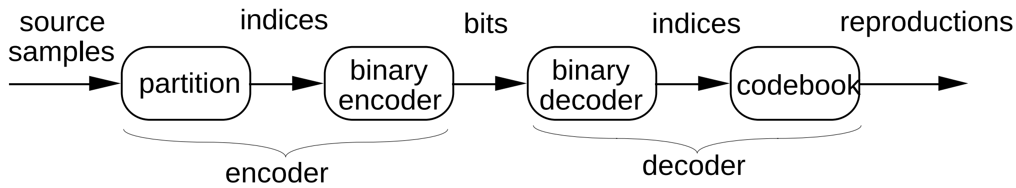
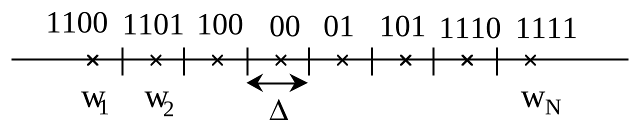
VARIABLE-LENGTH VQ (VL-VQ)

(a.k.a. Variable-Rate VQ, or VQ with Entropy Coding)

- **Variable-Length** = **Quantization** + **Lossless Variable-Length Binary Coding**



- **Range of options -- from simple to complex**



A. Uniform scalar quant with variable-length coding, one index at a time.

B. Nonuniform scalar quant w. variable-length coding, one index at a time.

C. Scalar quantization with higher-order variable-length coding -- either block coding of n indices at a time or nth-order conditional coding of the indices.

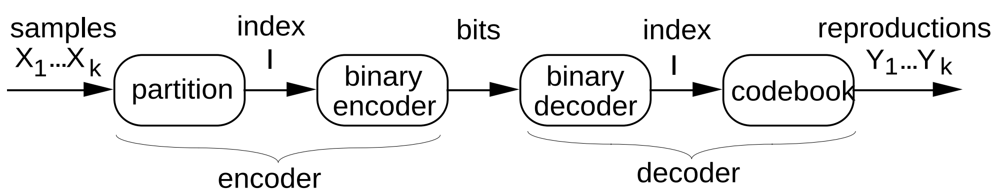
D. k-dimensional quant, with first-order variable-length coding, one index at a time.

E. k-dimen'l quant. with higher-order variable-length coding -- either block coding of n indices at a time, or nth-order condition'l coding of the indices.

G. k-dimen'l VQ with other types of lossless coding

- **Our focus:** Mainly on E. with block coding. A.-D. are special cases of E. Conditional coding is just a slight variation of E.

D. VL-VQ WITH 1ST-ORDER LOSSLESS CODING



- Losslessly encode one quantizer index at a time.
- Key characteristics:

  size: M (can be infinite)

  k-dim'l quantizing partition: $S = \{S_1, ..., S_M\}$,

  k-dim'l reprodution codebook: $C = \{\underline{w}_1, ..., \underline{w}_M\}$,

  binary prefix codebook: $C_b = \{\underline{c}_1, ..., \underline{c}_M\}$ with lengths $\{L_1,...,L_M\}$

- Encoding rule:

  $\underline{z} = \alpha(\underline{x}) = \underline{c}_i$ when $\underline{x} \in S_i$

- Decoding rule:

  $\underline{y} = \beta(\underline{z}) = \underline{w}_i$ when $\underline{z} = \underline{c}_i$

Note: because the codebook is prefix, there is no question about which bits are included in $\underline{z}$.

- Decompose encoder into "partitioning" and "binary encoding":

  Given $\underline{x}$, the partition produces index i when $\underline{x} \in S_i$
  The binary encoder outputs binary codeword $\underline{c}_i$ with length l_i

- Decompose decoder into "binary decoding" and "codevector lookup":

  The binary decoder decodes the bits into the index i.
  The codebook outputs $\underline{w}_i$.

- Quantization rule: describes the action of the code from source vector $\underline{X}$ to reproduction $\underline{Y}$:

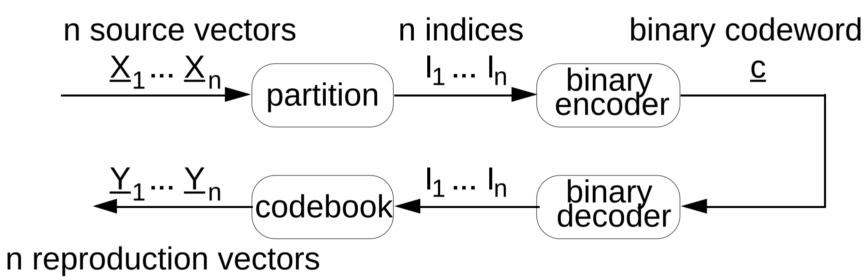
$$Q(\underline{x}) = \underline{w}_i \text{ when } \underline{x} \in S_i$$

- Rate = $R = \frac{1}{k} \sum_{\underline{x}} L(\underline{x})\, p(\underline{x}) = \frac{1}{k} \times$ rate of binary encoder

  Often assume: $R = \frac{1}{k} H(I) = \frac{1}{k} H(Q(X))$ (hence often called "VQ with entropy coding")

- Distortion = $D = E \frac{1}{k} ||\underline{X}-Q(\underline{X})||^2$ (not affected by choice of lossless coder)

E. VL-VQ WITH BLOCK LOSSLESS CODING



- Losslessly block encode n quantization indices at a time
- Key characteristics:
  - size: M (can be infinite)
  - k-dim'l quantizing partition: $S = \{S_1, ..., S_M\}$,
  - k-dim'l reprodution codebook of codevectors: $C = \{\underline{w}_1, ..., \underline{w}_M\}$,
  - n = "order" of binary prefix code, i.e. its input blocklength
  - binary prefix codebook: $C_b = \{\underline{c}_{\underline{i}} : \underline{i} \in \underline{\mathcal{I}}\}$ = binary prefix codebook
    - one codeword for each seq. of n indices in

$$\underline{\mathcal{I}} = \text{ set of index n-tuples } = \left\{\underline{i} = (i_1,\ldots,i_n) : 1 \le i_1 \le M, \ldots, 1 \le i_n \le M\right\}$$

$$\underline{c}_{\underline{i}} = (c_{\underline{i},1},\ldots,c_{\underline{i},l_{\underline{i}}}) = \text{ binary codeword of length } L_{\underline{i}} \text{ for } \underline{i} = (i_1,...,i_n)$$

- Encoding rule:

$$\underline{z} = \alpha(\underline{x}_1,\ldots,\underline{x}_n) = \underline{c}_{i_1,\ldots,i_n} \quad \text{when } \underline{x}_1 \in S_{i_1}, \underline{x}_2 \in S_{i_2}, \ldots, \underline{x}_n \in S_{i_n}$$

where $\underline{x}_1 = (x_1,\ldots,x_k)$, $\underline{x}_2 = (x_{k+1},\ldots,x_{2k})$, ..., $\underline{x}_n = (x_{nk-k+1},\ldots,x_{nk})$

- Decoding rule:

$$(\underline{y}_1,\ldots,\underline{y}_n) = \beta(\underline{z}) = (\underline{w}_{i_1},\ldots,\underline{w}_{i_n}) \quad \text{when } \underline{z} = \underline{c}_{i_1,\ldots,i_n}$$

Because the codebook is prefix, there is no question about which bits are included in $\underline{z}$.

- Decompose encoder into "partitioning" and "binary encoding":

"Partition" n successive k-dim'l source vectors $\underline{X}_1,\ldots,\underline{X}_n$ into indices $I_1,\ldots,I_n$, where $\underline{X}_j = (X_{j,1},\ldots,X_{j,k})$.

Losslessly encode n indices at once, $(I_1,\ldots,I_n)$, using FVL block lossless code with prefix codebook C_b having M^n binary codewords. ($M = \infty$ is possible.)

- Decompose decoder into "binary decoding" and "codevector lookup":

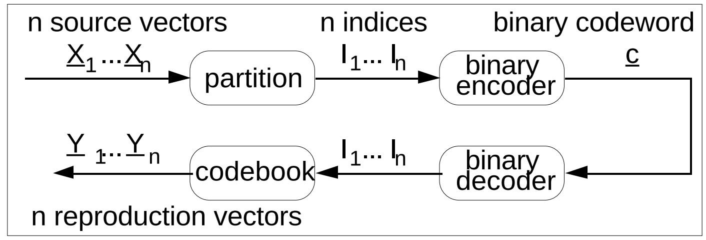
Decode binary codeword into n indices $I_1,\ldots,I_n$. Output corresponding quantization vectors $\underline{w}_{I_1},\ldots,\underline{w}_{I_n}$ as reproductions of $\underline{X}_1,\ldots,\underline{X}_n$, respectively.

- Quantization rule: same as before

$$Q(\underline{x}_i) = \underline{w}_i \quad \text{when } \underline{x}_i \in S_i$$

- "Block lossless binary coding" is an easy to analyze paradigm for studying the benefits of variable-length coding (a.k.a. entropy coding).
- Unless stated otherwise we assume source is stationary, so that $\underline{X}_j$ has the same pdf for all j, which will be denoted $f_{\underline{X}}(\underline{x})$, $f(\underline{x})$ or $f_k(\underline{x})$.

SUMMARY OF VL-VQ CHARACTERISTICS



Quantizer:

- k = dimension
- M = size (might be ∞)
- S = $\{S_1,\ldots,S_M\}$
  = k-dimensional partition
- C = $\{\underline{w}_1,\ldots,\underline{w}_M\}$ = k-dimensional codebook of codevectors

Binary Encoder:

- n = order of binary encoder (i.e. input blocklength)
- $C_b = \{\underline{c}_{\underline{i}} : \underline{i} \in \mathcal{I}\}$ = binary prefix codebook, one codeword for each seq. of n indices

  where $\mathcal{I}$ = set of cell index n-tuples = $\left\{\underline{i} = (i_1,\ldots,i_n) : 1 \le i_1 \le M, \ldots, 1 \le i_n \le M\right\}$

  $\underline{c}_{\underline{i}} = (c_{\underline{i},1},\ldots,c_{\underline{i},L_{\underline{i}}})$ = binary codeword of length $L_{\underline{i}}$ for $\underline{i}$

Derivative Characteristics:

- quantization rule: $Q(\underline{x}_j) = \underline{w}_i$ when $\underline{x}_j \in S_i$
- encoding rule: $\alpha(\underline{x}_1,\ldots,\underline{x}_n) = \underline{c}_{\underline{i}}$, when $\underline{i} = (i_1,\ldots,i_n)$ and $\underline{x}_j \in S_{i_j}$, j = 1,...,n
- decoding rule: $\beta(\underline{c}_{\underline{i}}) = (\underline{w}_{i_1}, \underline{w}_{i_2}, \ldots, \underline{w}_{i_n})$

PERFORMANCE

Distortion (by stationarity same as usual)

$$D = \frac{1}{k} E\,||\underline{X} - Q(\underline{X})||^2 = \frac{1}{k}\sum_{i=1}^{M} \int_{S_i} ||\underline{x} - \underline{y}_i||^2 f_k(\underline{x})\, d\underline{x}$$

where $\underline{X} = (X_1,\ldots,X_k)$ and $f_k(\underline{x})$ = pdf. Dist'n depends on S and C but not C_b.

Rate

$$R = \frac{1}{kn}\overline{L} = \frac{1}{kn}\sum_{\underline{i}} P_{\underline{i}} L_{\underline{i}} \text{ bits/sample}$$

where $\overline{L}$ = average length of binary codewords

$P_{\underline{i}}$ = probability of binary codeword $\underline{c}_{\underline{i}}$ = $\Pr(\underline{X}_1 \in S_{i_1},\ldots,\underline{X}_n \in S_{i_n})$, $\underline{i} \in \mathcal{I}$

Note: The size M of the quantizer has no direct relation to its rate.

- From lossless coding theorem

$$H(\underline{I}) \le \overline{L}^*_n \le H(\underline{I}) + 1$$

where $\overline{L}^*_n$ = least avg. length of prefix code for given VQ & n

$$H(\underline{I}) = -\sum_{\underline{i}} P_{\underline{i}} \log_2 P_{\underline{i}} \;=\; \text{entropy of } \underline{I} \quad (\text{or of } (\underline{Y}_1,\ldots,\underline{Y}_n))$$

- When analyzing VL-VQ we often assume assume the prefix code has been designed to have smallest possible average length, and as an approximation to the rate of the resulting VL-VQ, we assume

$$R = \frac{1}{kn} H(\underline{I}) = \frac{1}{k} H_n(I) = \frac{1}{kn} H(\underline{Y}_1,\ldots,\underline{Y}_n) = H_{kn}(Y)$$

- Note that that the rate is "nth"-order entropy in terms of the indices, but "kn" th-order entropy in terms of the reproduction Y.

IMPLEMENTATION AND COMPLEXITY

- Quantizer -- same issues as with fixed-length coding
- Lossless Coder -- table lookup is the brute force method
  + Table stores M^n binary codewords of various lengths
  + $M = 2^{kR_f}$ where $R_f = \frac{1}{k}\log_2 M$ is "fixed-length" rate
  + Complexity of brute force implementation of the lossless coder increases exponentially with exponent nkR_f.

OPTIMAL PERFORMANCE

As usual, we are interested in finding the best possible performance of codes of this type. Thus we are interested in the OPTA functions defined below. We are also interested in finding methods to design quantizers that attain the best possible performance specified by the OPTA functions.

- OPTA functions: For $k \geq 1$, $n \geq 1$, $R \geq 0$, and a given random vector $\underline{X}$ to quantize

  $\delta_{\underline{X}}(k,n,R) \triangleq$ least MSE of k-dim'l VQ w. nth-order entr'y coding & rate R or less

  $S_{\underline{X}}(k,n,R) \triangleq$ max SNR of k-dim'l VQ's w. nth-order entr'y coding & rate R or less

Notes:

- Often we omit the subsript $\underline{X}$.
- $\delta(k,n,0) = \sigma^2$ for every k,n.
- For every k,n, $\delta(k,n,R)$ decreases as $R \to \infty$ to zero.
- Unlike the staircase form of the fixed-length OPTA $\delta(k,R)$, the variable-length OPTA $\delta(k,n,R)$ is a continuous function of R.
- Though $\delta(k,n,R)$ is often a convex $\cup$ function of R, there are examples where it is not.
- For any k,n,R there is a quantizer with rate R or less and MSE exactly equal to $\delta(k,n,R)$.

- For a random process source, we are also interested in

  $\delta(R) \triangleq \inf_{k,n} \delta(k,n,R)$ = least MSE of VL-VQ with rate R or less, and any k,n

  $S(R) \triangleq \sup_{k,n} \delta(k,n,R)$ = max SNR of VQ with EC & rate R or less, and any k,n

OPTIMALITY CONDITIONS FOR VARIABLE-LENGTH VQ WITH 1ST-ORDER LOSSLESS CODING

There are three principal characteristics to be optimized: the partition, binary codebook and reproduction codebook. We discuss them in reverse order.

Optimal reproduction codebook for a given partition

Suppose we are given a k-dimensional partition $S = \{S_1,\ldots,S_M\}$ of size M and a binary codebook $C_b = \{\underline{c}_1,\ldots,\underline{c}_M\}$ for quantizing a random vector $\underline{X}$, and suppose we seek the best k-dimensional reproduction codebook $C = \{\underline{w}_1,\ldots,\underline{w}_M\}$ to use with this partition and binary codebook. Since the reproduction codebook does not affect the rate, we should choose it to minimize MSE. Thus the optimal codebook is found in exactly the same way as for fixed-length quantization

Optimal Codebook Condition:

Given a partition $S = \{S_1, \ldots,S_M\}$ and source density $f_{\underline{X}}(\underline{x})$, the unique codevectors that minimize MSE are the "centroids"

$$\underline{w}_i = E[\underline{X}|\underline{X}\in S_i] = \int \underline{x}\, f_{\underline{X}}(\underline{x}|\underline{X}\in S_i)\, d\underline{x} \;, \quad i = 1,\ldots,M \qquad (**)$$

where

$$f_{\underline{X}}(\underline{x}|\underline{X}\in S_i) = \begin{cases} \dfrac{f_{\underline{X}}(\underline{x})}{\Pr(\underline{X}\in S_i)}, & \underline{x}\in S_i \\ 0 , & \text{else} \end{cases}$$

Notes:

- There is one and only one optimal codebook for a given partition.
- The optimal codebook does not depend on the binary codebook.

Optimal binary codebook for given partition

Suppose we are given a k-dimensional partition $S = \{S_1,\ldots,S_M\}$ of size M and a k-dimensional reproduction codebook $C = \{\underline{w}_1,\ldots,\underline{w}_M\}$ for quantizing a random vector $\underline{X}$, and suppose we seek the best binary codebook $C_b = \{\underline{c}_1,\ldots,\underline{c}_M\}$ to use with this partition and reproduction codebook. Since the binary codebook does not affect the distortion, we should choose it to minimize the rate. Thus, it can be designed using Huffman's algorithm applied to $\{p_1,\ldots,p_M\}$ where

$$p_i = \Pr(\underline{X}\in S_i).$$

As mentioned earlier, this results in average length $\overline{L}^*$

$$H(I) \le \overline{L}^* \le H(I) + 1$$

where

$$H(I) = -\sum_{i=1}^{M} p_i \log_2 p_i$$

Optimal partition for given reproduction and binary codebooks

Suppose we are given a k-dimensional reproduction codebook $C = \{\underline{w}_1,\ldots,\underline{w}_M\}$ of size M, and a binary codebook $C_b = \{\underline{c}_1,\ldots,\underline{c}_M\}$ with lengths $\{L_1,\ldots,L_M\}$ for quantizing a random vector $\underline{X}$, and suppose we seek the best k-dimensional partition $S = \{S_1,\ldots,S_M\}$ to use with this codebook and binary codebook. Because the partition affects both the distortion and the rate, optimizing it is not so simple. We are used to having the partition affect the distortion, but since the partition affects the probability of the binary codewords, it also affects the rate.

Because the partition affects the rate, the choice of best partition will depend on the rate that we wish to attain. Specifically, given a target rate R, we should find the partition that minimizes MSE among all those that cause the rate to be R or less. This is a constrained minimization:

$$\min_{S:\, r(S)\le R} d(S)$$

where

$$d(S) = \frac{1}{k} E\,\|\underline{X}-Q_{S,C}(\underline{X})\|^2,$$

$Q_{S,C}$ denotes the quantization rule induced by S and C, and

$$r(S) = \frac{1}{k}\sum_{i=1}^{M} p_i L_i ,$$

where $p_i = \Pr(\underline{X}\in S_i)$.

To obtain an optimality condition for the partition, we use a Lagrange multiplier approach. Specifically, we hypothesize a positive number μ, called the *Lagrange multiplier*, and find a condition on S that guarantees that it minimizes

$$d(S) + \mu\, r(S) .$$

Lagrange multiplier theory then indicates that there will be some value of μ such that the partition that minimizes $d(S) + \mu\, r(S)$ will have rate R and has smallest distortion among all quantizers with rate R or less.

Accordingly, let us assume $\mu>0$ is fixed, and see what condition S should satisify in order minimize $d(S) + \mu\, r(S)$. We begin by observing that

$$d(S) + \mu\, r(S) = \frac{1}{k}\sum_{i=1}^{M}\int_{S_i} \|\underline{x}-\underline{w}_i\|^2 f(\underline{x})\, d\underline{x} + \mu\, \frac{1}{k}\sum_{i=1}^{M} p_i L_i$$

$$= \frac{1}{k}\sum_{i=1}^{M}\int_{S_i} \|\underline{x}-\underline{w}_i\|^2 f(\underline{x})\, d\underline{x} + \mu\, \frac{1}{k}\sum_{i=1}^{M} L_i \int_{S_i} f(\underline{x})\, d\underline{x}$$

$$= \frac{1}{k}\sum_{i=1}^{M}\int_{S_i} \left(\|\underline{x}-\underline{w}_i\|^2 + \mu L_i\right) f(\underline{x})\, d\underline{x}$$

Now we see that to minimze $d(S) + \mu\, r(S)$, we should put $\underline{x}$ in the cell S_i for which $\left(\|\underline{x}-\underline{w}_i\|^2 + \mu L_i\right)$ is smallest. That is,

$$S_i = \left\{\underline{x} : \|\underline{x}-\underline{w}_i\|^2 + \mu L_i \le \|\underline{x}-\underline{w}_j\|^2 + \mu L_j, \text{ for all } j\right\}$$

We summarize:

Optimal Partition Condition:

The optimal partition for reproduction codebook $C = \{\underline{w}_1,\dots,\underline{w}_M\}$, and binary codebook $C_b = \{\underline{c}_1,\dots,\underline{c}_M\}$ with lengths $\{L_1,\dots,L_M\}$ for quantizing a random vector $\underline{X}$ at rate R or less has cells

$$S_i = \left\{ \underline{x} : \ \|\underline{x}-\underline{w}_i\|^2 + \mu L_i \le \|\underline{x}-\underline{w}_j\|^2 + \mu L_j, \text{ for all } j \right\}, \ i = 1,\dots,M$$

for some suitable choice of $\mu > 0$.

Notes:

- If $\mu = 0$, then the optimal partition is just a Voronoi partition, i.e. the partition that minimizes distortion. It does not take rate into account.
- If μ is very large, then the optimal partition minimizes rate and ignores distortion. It puts essentially all $\underline{x}$'s into the cell (or cells) that are assigned the shortest binary codeword(s). Thus the average length is, approximately, the length of the shortest binary codeword.
- For intermediate values of μ, the opt'l partition is a compromise between the Voronoi partition that minimizes distortion and the partition of the previous bullet that minimizes rate. In comparison to the Voronoi partition, it induces larger distortion and less rate. The cells assigned smaller lengths L_i are larger than the corresponding cells in the Voronoi partition.

  Why? The perpendicular bisector between $\underline{w}_i$ and $\underline{w}_j$ moves toward the codevector assigned a larger binary codeword length, making the cell corresponding to this codevector smaller.

LLOYD-STYLE VL-VQ DESIGN ALGORITHM

The optimality conditions just discussed suggest the following algorithm for designing a VL-VQ. Assume we are given k, M, the pdf $f(\underline{x})$ of $\underline{X}$ and a target rate R.

1. Make an initial choice of the reproduction codebook C, binary codebook C_b and Lagrange multiplier μ.

2. Find the best partition S for the current C, C_b and μ:

$$S_i = \left\{ \underline{x} : \ \|\underline{x}-\underline{w}_i\|^2 + \mu L_i \le \|\underline{x}-\underline{w}_j\|^2 + \mu L_j, \text{ for all } j \right\}, \ i = 1,\dots,M$$

3. Find the best reproduction codebook C for the given partition S:

$$\underline{w}_i = E[\underline{X} | \underline{X} \in S_i], \ i = 1,\dots,M$$

4. Find the best binary codebook C_b for the given partition S by applying Huffman's algorithm to $\{p_1,\dots,p_M\}$, where $p_i = \Pr(\underline{X} \in S_i)$.

5. Test to see if the codebook or the distortion has changed significantly. If so, go to Step 2. If not, go to Step 6.

6. Compute the rate.

7. If the rate is close enough to R, then terminate the algorithm. If not, then change μ to $\mu + \alpha(\text{rate}-R)$, where α is a small number, and return to Step 2.

Notes:

- As with fixed-length VQ, this algorithm is usually run with a training sequence rather than an explicit formula for $f(\underline{x})$.
- As a simplification, the algorithm is usually run without actually implementing the Huffman algorithm in Step 4. Instead, the algorithm assumes

$$L_i = -\log_2 p_i$$

  and it principally iterates Steps 2 and 3. After it terminates, one runs the Huffman algorithm on the final cell probabilities $p_1,\ldots,p_M$.

  VL-VQ's designed in this way are often called "entropy-coded VQ" or "entropy-contrained VQ" because they're designed to minimize distortion vs. entropy.

- When performing Step 2, it can happen that some partition cells wind up empty. This is because an optimal VL-VQ quantizer with rate R might have fewer than M cells. Since a priori one doesn't know what M is best for a given R, one usually begins with a large M and allows the algorithm to reduce the number of cells below M.
- VL-VQ design algorithms are generally more challenging to create than FL-VQ design algorithms, because of the needs to adjust μ, to find M, and for other reasons.
- On the other hand, high-resolution analysis will indicate a universally good way to design quantizers, without an iterative algorithm

QUANTIZER TRANSFORMATIONS

- One can scale, shift and transform VL-VQ's just as easily as FL-VQ's.
- The opta functions change in the same way as for FL-VQ's.

  For example, if $\underline{Y} = a\,\underline{X}$, then $\delta_{\underline{Y}}(k,n,R) = a^2\,\delta_{\underline{X}}(k,n,R)$

QUANTIZER COMBINATIONS

- One can combine VL-VQ's using the product, two-stage, and multistage approaches, just as with FL-VQ's.
- The union combination of VL-VQ's works, provided one uses encoding rule b. on p. 79 of the VQ notes. Encoding rule a. would need some modification.

HIGH-RESOLUTION ANALYSIS

- As before, there are two goals:

  To find an approximate formula or formulas that analyze the performance of quantizers in terms of the pdf and key gross characteristics.

  To find an approximate formula for the opta function $\delta(k,n,R)$ for large R.

- As before, we assume the VQ has mostly small cells, negligible overload distortion, neighboring cells with similar sizes & shapes, and inertial profile $m(\underline{x})$, approximately.

- We also assume it can be said to "have a point density". However, since quantizer size is unimportant (e.g. not closely related to rate), we use "unnormalized point" density $\Lambda(\underline{x})$, which is a function such that

1. $\int_A \Lambda(\underline{x})\, d\underline{x} \cong$ number of codevectors (or cells) in region A
2. If A is small, but much larger than the cells in the vicinity of $\underline{x}$,

   $\Lambda(\underline{x})\, |A| \cong$ # points/cells in A
3. $\Lambda(\underline{x}) \geq 0$, $\int \Lambda(\underline{x})\, d\underline{x} = M =$ total number of quantization points (can be ∞)
4. Ordinarily $\Lambda(\underline{x})$ is a smooth or piecewise smooth function.
5. $\Lambda(\underline{x}) \cong \frac{1}{|S_i|}$ when $\underline{x} \in S_i$

DISTORTION: BENNETT'S INTEGRAL

Since distortion is not influenced by the binary coder, the same Bennett's itnegral applies. That is,

$$D \cong \int \frac{m(\underline{x})}{\Lambda^{2/k}(\underline{x})} f_k(\underline{x})\, d\underline{x}$$

RATE: ASYMPTOTIC ENTROPY FORMULA FOR VL-VQ

To analyze rate, we assume that it equals the entropy of the quantizer output, suitably normalized. Our approximate formula for rate is given below.

Fact: If $\underline{X}_1,\ldots,\underline{X}_n$ are identically distributed k-dimensional vectors, then under high-resolution conditions,

$$R \cong \frac{1}{kn} H(\underline{I}) \cong h_{kn} + \frac{1}{k} \int f_k(\underline{x}) \log_2 \Lambda(\underline{x}) \, d\underline{x}$$

where

$\underline{I} = (I_1,\ldots,I_n)$, where I_j = index resulting from quantizing $\underline{X}_j$

$\underline{x} = (x_1,\ldots,x_k)$

$h_{kn} = \frac{1}{kn} h(X_1,\ldots,X_{kn})$ = kn-th order <u>differential entropy</u>

$$= -\frac{1}{kn} \int f_{kn}(x_1\ldots x_{kn}) \log_2 f_{kn}(x_1\ldots x_{kn}) \, dx_1\ldots dx_{kn}$$

We'll derive this shortly.

Note: *Differential entropy* is not the same as *entropy* [1]. For a continuous random variable or vector, entropy = ∞, whereas differential entropy can be finite.

[1]Sometimes people. who consider differential entropy in a context in which entropy does not appear. call it "entropy". But we do not have this luxury.

Most Important Example: *(later we learn why this example is so important)*

For a uniform scalar quantizer with step size Δ and infinitely many levels

$$\Lambda(x) \cong \frac{1}{\Delta}$$

Then using the approximate rate formula with this Λ and k=1 we find

$$R \cong h_{kn} + \frac{1}{k} \int f_k(\underline{x}) \log_2 \Lambda(\underline{x}) \, d\underline{x}$$

$$= h_n - \log \Delta = h_n - \frac{1}{2} \log 12 \frac{\Delta^2}{12}$$

$$\cong h_n - \frac{1}{2} \log 12 D$$

Equivalently

$$D \cong \frac{1}{12} 2^{2h_n} 2^{-2R}$$

This formula for D also applies when there are only finitely many levels, provided the overload distortion is small.

DERIVATION OF ASYMPTOTIC FORMULA FOR $H(\underline{I})$

First case: $n = 1$ (for simplicity)

$$H(I) = -\sum_i P_i \log P_i \,, \qquad \text{where } P_i = \Pr(\underline{X}_1 \in S_i) = \int_{S_i} f_k(\underline{x})\, d\underline{x}$$

$$= -\sum_i \left(\int_{S_i} f_k(\underline{x})\, d\underline{x} \right) \log \left(\int_{S_i} f_k(\underline{x})\, d\underline{x} \right)$$

$$\cong -\sum_i \big(f_k(\underline{w}_i)\, |S_i| \big) \log \big(f_k(\underline{w}_i)\, |S_i| \big) \qquad \text{because cells are small}$$

$$= -\sum_i \big(f_k(\underline{w}_i) \log f_k(\underline{w}_i) \big) |S_i| \; - \; \sum_i \left(f_k(\underline{w}_i) \log \frac{1}{\Lambda(\underline{w}_i)} \right) |S_i|$$

recall $\Lambda(\underline{x}) \cong \frac{1}{|S_i|}$ when $\underline{x} \in S_i$

$$\cong -\int f_k(\underline{x}) \log f_k(\underline{x})\, d\underline{x} + \int f_k(\underline{x}) \log \Lambda(\underline{x})\, d\underline{x}$$

$$= k\, h_k + \int f_k(\underline{x}) \log \Lambda(\underline{x})\, d\underline{x}$$

General case: $n \geq 1$

$$H(\underline{I}) = -\sum_{\underline{i}} P_{\underline{i}} \log P_{\underline{i}}$$

$$= -\sum_{\underline{i}} \left(\int_{S_{\underline{i}}} f_{kn}(\underline{x})\, d\underline{x} \right) \log \left(\int_{S_{\underline{i}}} f_{kn}(\underline{x})\, d\underline{x} \right), \qquad \text{where } \underline{x} = (x_1 \ldots x_{kn}),\; S_{\underline{i}} = (S_{i_1} \times S_{i_2} \times \ldots \times S_{i_n})$$

$$\cong -\sum_{\underline{i}} \big(f_{kn}(\underline{w}_{\underline{i}})\, |S_{\underline{i}}| \big) \log \big(f_{kn}(\underline{w}_{\underline{i}})\, |S_{\underline{i}}| \big), \qquad \text{where } \underline{w}_{\underline{i}} = (\underline{w}_{i_1}, \underline{w}_{i_2}, \ldots, \underline{w}_{i_n})$$

$$= -\sum_{\underline{i}} \big(f_{kn}(\underline{w}_{\underline{i}}) \log f_{kn}(\underline{w}_{\underline{i}}) \big) |S_{\underline{i}}| \; - \; \sum_{\underline{i}} \big(f_{kn}(\underline{w}_{\underline{i}}) \log |S_{\underline{i}}| \big) |S_{\underline{i}}|$$

The first summation in the above can be approximated by the integral

$$-\int f_{kn}(\underline{x}) \log f_{kn}(\underline{x})\, d\underline{x} = kn\, h_{kn} \qquad (*)$$

Before approximating the second sum, note that

$$\log |S_i| = \log\left(|S_{i_1}|\,|S_{i_2}| \ldots |S_{i_n}|\right) = \sum_{j=1}^{n} \log |S_{i_j}| \cong -\sum_{j=1}^{n} \log \Lambda(\underline{w}_{i_j})$$

Substitute the above into the second summation:

$$-\sum_{\underline{i}} \left(f_{kn}(\underline{w}_{\underline{i}}) \log |S_{\underline{i}}|\right) |S_{\underline{i}}|$$

$$= -\sum_{\underline{i}} \left(f_{kn}(\underline{w}_{\underline{i}}) \left(-\sum_{j=1}^{n} \log \Lambda(\underline{w}_{i_j})\right)\right) |S_{\underline{i}}|$$

$$\cong \int f_{kn}(\underline{x}_1 \ldots \underline{x}_n) \sum_{j=1}^{n} \log \Lambda(\underline{x}_j)\; d\underline{x}_1 \ldots d\underline{x}_k$$

$$= \sum_{j=1}^{n} \int f_{kn}(\underline{x}_1 \ldots \underline{x}_n) \log \Lambda(\underline{x}_j)\; d\underline{x}_1 \ldots d\underline{x}_k$$

$$= \sum_{j=1}^{n} \int f_k(\underline{x}_j) \log \Lambda(\underline{x}_j)\; d\underline{x}_j$$

$$= n \int f_k(\underline{x}_1) \log \Lambda(\underline{x}_1)\, d\underline{x}_1 \quad \text{because } \underline{X}_1,\ldots,\underline{X}_n \text{ are identical} \qquad (**)$$

Substituting (*) and (**) into the expression for $H(\underline{I})$ gives

$$\frac{1}{kn} H(\underline{I}) \cong \frac{1}{kn}\left(kn\, h_{kn} + n \int f_k(\underline{x}_1) \log \Lambda(\underline{x})\, d\underline{x}_1\right)$$

$$= h_{kn} + \frac{1}{k} \int f_k(\underline{x}) \log_2 \Lambda(\underline{x})\, d\underline{x}$$

ZADOR THEOREM FOR VARIABLE-LENGTH VQ

For a stationary source and large R, the least distortion of k-dim'l VQ with nth-order entropy coding and rate R or less is

$$\delta(k,n,R) \cong \mathcal{Z}(k,n,R) \triangleq m_k^v \sigma^2 \eta_{kn} 2^{-2R}$$

where

$\mathcal{Z}(k,n,R)$ = Zador function for k-dim'l VQ with nth-order entropy coding

m_k^v = best inert'l profile for variable-length coding

$$\eta_k = \frac{1}{\sigma^2} 2^{2h_k}, \quad h_k = \frac{1}{k} h(X_1,\ldots,X_k)$$

$$\sigma^2 = \frac{1}{k}\sum_{i=1}^{k} \mathrm{var}(X_i)$$

Equivalently,

$$S(k,n,R) \cong 6.02\, R - 10 \log_{10} m_k^v \eta_{kn} \;.$$ (Again, 6 dB per bit.)

Notes:

- k-dim'l VQ-EC with n = 1 is at least as good as k-dim'l VQ-FL, because the latter is a special case of the former. Therefore, $\delta(k,n,R) \le \delta(k,R)$
- Later we show directly that $\eta_k \le \beta_k$, which implies $\mathcal{Z}(k,1,R) \le \mathcal{Z}(k,R)$.

- The proof shows that an optimal k-dimensional quantizer with nth-order variable-length coding has, approximately, a constant point density, namely,

$$\Lambda(\underline{x}) = \Lambda_k^* = 2^{k(R-h_{kn})}$$

constant inertial profile, namely,

$$m(\underline{x}) = m_k^v$$

and distortion

$$D \cong m_k^v (\Lambda_k^*)^{-2/k} = Z(k,nR).$$

- It follows that for large R, a uniform scalar quantizer is an optimal VL scalar quantizer. Because of this, there is no reason to design scalar quantizer with an iterative algorithm, unless R is small.
- Similarly, when R is large, an optimal k-dimensional VL quantizer has constant point density and inertial profile. Therefore, its partition, could be a lattice, tesselation, or periodic tesselation.
- Most people think

$$m_k^v = m_k^*$$

but this has never been proved or disproved, except that it is easily seen to be true for $k = 1$. From now on, we will simply asume it is true.

Proof of Zador Theorem

We begin with

$$\delta(k,n,R) \cong \min_{m(\underline{x}),\Lambda(\underline{x})} \int \frac{m(\underline{x})}{\Lambda^{2/k}(\underline{x})} f(\underline{x})\, d\underline{x}$$

where $\underline{x} = (x_1 \ldots x_k)$ and the minimization is over all inertial profiles $m(\underline{x})$ and all point densities $\Lambda(\underline{x})$ such that

$$h_{kn} + \frac{1}{k}\int f(\underline{x}) \log \Lambda(\underline{x})\, d\underline{x} \le R \qquad (\text{such a } \Lambda \Rightarrow \text{rate} \lesssim R)$$

- Best inertial profile: (recall the assumption $m_k^v = m_k^*$)

For same reason as with fixed-length quantization, the best inertial profile is

$$m_k^*(\underline{x}) = m_k^* \triangleq \min_{\substack{\text{valid k-dimensional} \\ \text{inertial profiles m}}} \min_{\underline{x}} m(\underline{x})$$

- It follows that

$$\delta(k,n,R) \cong m_k^* \min_{\Lambda(\underline{x})} \int \frac{1}{\Lambda^{2/k}(\underline{x})} f(\underline{x})\, d\underline{x}$$

where the min is again taken over functions $\Lambda(\underline{x})$ such that $\Lambda(\underline{x}) \ge 0$ and

$$h_{kn} + \frac{1}{k}\int f(\underline{x}) \log \Lambda(\underline{x})\, d\underline{x} \le R$$

- Best point density:

Suppose $\Lambda(\underline{x})$ satisfies

$$h_{kn} + \frac{1}{k}\int f(\underline{x}) \log_2 \Lambda(\underline{x})\, d\underline{x} \le R \qquad (*)$$

Then by convexity of the logarithm and Jensen's inequality (see next page)

$$\log_2 \int \frac{1}{\Lambda^{2/k}(\underline{x})} f(\underline{x})\, d\underline{x} \ge \int \log_2\left(\frac{1}{\Lambda^{2/k}(\underline{x})}\right) f(\underline{x})\, d\underline{x}$$

equality iff $\Lambda(\underline{x})$ is constant with probability one

$$= -\frac{2}{k}\int f(\underline{x}) \log \Lambda(\underline{x})\, d\underline{x} \ge 2h_{kn} - 2R$$

Hence,

$$\int \frac{1}{\Lambda^{2/k}(\underline{x})} f(\underline{x})\, d\underline{x} \ge 2^{2h_{kn}} 2^{-2R} \qquad (**)$$

with equality iff $\Lambda(\underline{x})$ is a constant with probability one.

$$(*) \text{ and } (**) \Rightarrow \delta(k,n,R) \cong m_k^* 2^{2h_{kn}} 2^{-2R} = m_k^* \sigma^2 \eta_{kn}\, 2^{-2R}$$

Moreover, we have shown that the optimal point density is a constant.
The constant must be such that $(*)$ holds with equality. Therefore,

$$\Lambda(\underline{x}) = 2^{k(R-h_{kn})} \triangleq \Lambda_k^*$$

- Jensen's inequality: Let $c(\underline{x})$ be convex $\cap$, and $d(\underline{x})$ be arbitrary, then

$$c\left(\int d(\underline{x}) f(\underline{x})\, d\underline{x}\right) \ge \int c(d(x))\, f(\underline{x})\, d\underline{x},$$

Equivalently, if $Y = d(\underline{X})$, then $c(E[Y]) \ge E[c(Y)]$.

If c is strictly convex, then equality holds if and only if $d(\underline{X})$ is constant with probability one.

Summary of Fixed- and Variable-Length VQ

Let 0th-order entropy coding (n=0) denote fixed-length coding.

Given a stationary source and large R, the least distortion of VQ with dimension k, nth-order entropy coding, and rate R or less is

$$\delta(k,n,R) \cong m_k^* \sigma^2 \alpha_{k,n} 2^{-2R} = \mathcal{Z}(k,n,R)$$

where

$$\alpha_{k,n} = \begin{cases} \beta_k , & n=0 \quad \text{(fixed-length coding)} \\ \eta_{kn} = \frac{1}{\sigma^2} 2^{2h_{kn}} , & n \ge 1 \quad \text{(nth-order VL coding)} \end{cases}$$

- **Notice** the "," in $\alpha_{k,n}$ but not in η_{kn} or h_{kn}!
- The best k-dimen'l VQ to use with fixed-length coding (n=0) has
  - point density $\lambda_k^*(\underline{x}) = c\, f^{k/(k+2)}(\underline{x})$
  - constant inertial profile $m(\underline{x}) = m_k^*$
- The best k-dimen'l VQ to use with variable-length binary coding (n≥1) has
  - constant point density
  - constant inertial profile $m(\underline{x}) = m_k^*$.

What Happens as k and n Change?

As usual, consider a stationary source.

Recall:

$$\delta(k,n,R) \cong \sigma^2 m_k^* \alpha_{k,n} 2^{-2R}$$

m_k^* decreases subadditively to $m_\infty^* = \frac{1}{2\pi e}$

$$\alpha_{k,n} = \begin{cases} \beta_k , & n = 0 \\ \frac{1}{\sigma^2} 2^{2h_{kn}} , & n \ge 1 \end{cases}$$

β_k decreases submultiplicatively to β_∞

2^{2h_k} decreases monotonically to 2^{2h_∞}

Therefore,

$\alpha_{k,0}$ decreases submultiplicatively with k to β_∞

$\alpha_{k,n}$ decreases monotonically with k to $2^{2h_\infty}/\sigma^2$ for $n \ge 1$

$\alpha_{k,n}$ decreases monotonically with n to $2^{2h_\infty}/\sigma^2$

Key Fact to be proved later: $2^{2h_\infty}/\sigma^2 = \beta_\infty$

Therefore,

$\alpha_{k,n}$ decreases (monoton'ly or submult'ly) with n or k to $\beta_\infty = 2^{2h_\infty}/\sigma^2$

CONCLUSIONS

1. The least distortion of vector quantization with rate R or less, with any dimension, and with fixed-length coding or with variable-length encoding of any order is

$$\delta(R) \cong \sigma^2 m^*_\infty \beta_\infty 2^{-2R}$$

Among other things, this says that the best possible performance with variable-length coding is no better than the best possible performance with fixed-length coding.

2. Increasing n with k fixed:

$\delta(k,n,R)$ decreases monotonically to the limit

$$\delta(k,\infty,R) \cong \sigma^2 m^*_k \beta_\infty 2^{-2R} \cong \frac{m^*_k}{m^*_\infty}\,\delta(R) \qquad \text{(space filling loss)}$$

> Therefore, for large n and arbitrary k,
>
> $$\delta(k,n,R) \cong \frac{m^*_k}{m^*_\infty}\,\delta(R)$$

Among other things, this shows that one needs large k in order to approach the best possible performance.

3. Increasing k with n fixed:

$\delta(k,n,R)$ "decreases", though not monotonically, to the limit

$$\delta(\infty,n,R) \cong \sigma^2 m^*_\infty \beta_\infty 2^{-2R} = \delta(R) \qquad \text{(no loss)}$$

> Therefore, for large k and arbitrary n (even $n = 0$),
>
> $$\delta(k,n,R) \cong \delta(R)$$

Among other things this indicates that for large k, increasing n does not improve the best possible performance attainable with for that k. That is, one can attain the best possible performance, even with $n = 0$ or 1.

4. To get the best possible performance we must have

(a) k large enough that $m^*_k/m^*_\infty \cong 1$,

i.e. well shaped cells,

(b) k and/or n large enough that $\alpha_{k,n} \cong \beta_\infty$,

i.e. good point density and good exploitation of memory.

Usually (b) is more important (a).

5. What's the point of variable-length coding if the best possible performance with variable-length coding is no better than the best possible performance without it?

From the point of view of VQ, the reason to use variable-length coding, instead of fixed-length coding) is to permit a VQ with smaller dimension, and consequently less complexity, to work well.

k=1, the extreme case, i.e. scalar quantization:

- with n = 0 (fixed-length coding)

$$\delta(1,0,R) \cong \sigma^2 m_1^* \beta_1 2^{-2R} = \frac{m_1^*}{m_\infty^*} \frac{\beta_1}{\beta_\infty} \delta(R)$$

- with large n (high-order variable-length coding)

$$\delta(1,\infty,R) \cong \sigma^2 m_1^* \beta_\infty 2^{-2R} = \frac{m_1^*}{m_\infty^*} \delta(R), \text{ where } \frac{m_1^*}{m_\infty^*} = 1.42 \text{ or } 1.53 \text{ dB}$$

- The variable-coding causes β_1 to be replaced by β_∞.
- Moreover, the best scalar quantizer for use with variable-length coding is a uniform scalar quantizer, which is the simplest quantizer.
- This shows that with variable-length coding, a uniform scalar quantizer can have performance within only 1.53 dB of the best VQ of any type!

6. What's the point of vector quantization if uniform scalar quantization plus variable-length coding can come within 1.53 dB of the best VQ of any type?

From the point of view of the binary code, the purpose of VQ is to permit a lower order variable-length coder to be used, i.e. it permits a simpler lossless coder. For example, if uniform scalar quantization were used, the variable-length coder must exploit the memory in the source, i.e. it would have to be complex.

VQ also reduces the space filling loss, i.e. it improves cell shapes.

7. It is worth re-emphasizing that one can attain:

- the best possible performance (D vs. R) by choosing k large and n = 0; i.e. with a complex quantizer and a simple fixed-length binary encoder.
- the best possible performance minus only 1.53 dB by choosing a uniform scalar quantizer and an entropy coder with large n;
  i.e. with a simple quantizer and a complex entroopy coder.

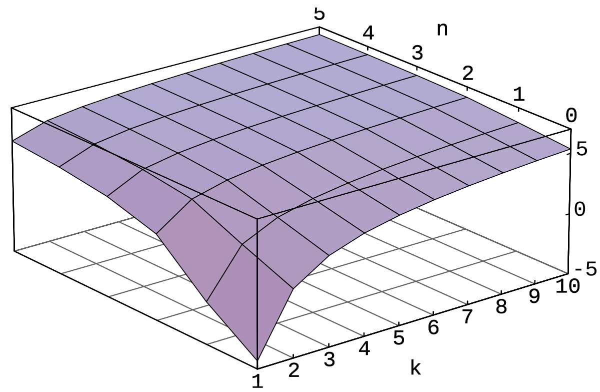
Which is simpler? Hard to say. Both are very complex. Good systems are usually compromises; i.e. a nontrivial quantizer and a nontrivial entropy coder.

In some applications variable-length coding is not an option, in which case fixed-length coding must be used.

EXAMPLE: GAUSS-MARKOV SOURCE, $\rho = .9$

$$S(k,n,R) \cong 6.02\,R - 10 \log_{10} m_k^* \alpha_{k,n}$$

Plot of $-\log_{10} \alpha_{k,n}$



PROPERTIES OF DIFFERENTIAL ENTROPY h AND THE ZADOR FACTOR η_K

Most are extensions of properties of the differential entropy of one random variable. Many proofs are similar to those of analogous properties for ordinary entropy.

Definitions:

$$h(X_1,\ldots,X_k) \triangleq -\int f(\underline{x}) \log_2 f(\underline{x})\, d\underline{x}$$

$$h_k \triangleq \frac{1}{k} h(X_1,\ldots,X_k) = \text{kth order differential entropy}$$

$$\eta_k \triangleq \frac{1}{\sigma^2} 2^{2h_k} = \text{Zador factor for VL-VQ}$$

$$h_k = \frac{1}{2} \log_2 \sigma^2 \eta_k$$

Note that h and h_k can be positive or negative!

(1) $h_k \le \frac{1}{2}\log_2 \sigma^2\beta_k$ and $\eta_k \le \beta_k$

where β_k is Zador's factor. Equality holds in each iff $f(\underline{x})$ has the same value wherever it is not zero, e.g. if it is uniform.

Derivation:

$$h_k = -\frac{1}{k}\int f(\underline{x}) \log_2 f(\underline{x})\, d\underline{x} = \frac{k+2}{2k}\int f(\underline{x}) \log_2 f^{-2/(k+2)}(\underline{x})\, d\underline{x}$$

$$= \frac{k+2}{2k}\, E \log_2 Y, \quad \text{where } Y = f^{-2/(k+2)}(\underline{X})$$

$$\le \frac{k+2}{2k} \log_2 EY \quad \text{by Jensen's ineq. } (\log_2 \text{ is concave})$$

$$= \frac{k+2}{2k} \log_2 \int f(\underline{x})\, f^{-2/(k+2)}(\underline{x})\, d\underline{x}$$

$$= \frac{1}{2}\log_2\left(\int f^{k/(k+2)}(\underline{x})\, d\underline{x}\right)^{(k+2)/k} = \frac{1}{2}\log_2 \sigma^2\beta$$

$$\Rightarrow \eta_k = \frac{1}{\sigma^2} 2^{2h_k} \le \beta_k$$

Since $\log_2$ is strictly concave, equality holds if and only if Y is constant w.p.1;

i.e. iff $\Pr(f^{-2/(k+2)}(\underline{X}) = c) = 1$ for some some c,

i.e. iff $f(\underline{x})$ has the same value wherever it is not zero.

(2) If $\underline{Y} = a\,\underline{X} + \underline{b}$, $a \ne 0$, then

$$h_{Y,k} = h_{X,k} + \log_2 |a| \quad \text{and} \quad \eta_{Y,k} = \eta_{X,k}$$

Derivation: Since $f_{\underline{Y}}(\underline{y}) = \frac{1}{|a|^k} f_{\underline{X}}(\frac{\underline{y}-\underline{b}}{a})$,

$$h_{Y,k} = -\frac{1}{k}\int f(\underline{y}) \log_2 f(\underline{y})\, d\underline{y} = -\frac{1}{k}\int \frac{1}{|a|^k} f_{\underline{X}}(\tfrac{\underline{y}-\underline{b}}{a}) \log_2\left(\frac{1}{|a|^k} f_{\underline{X}}(\tfrac{\underline{y}-\underline{b}}{a})\right) d\underline{y}$$

$$= -\frac{1}{k}\int \frac{1}{|a|^k} f_{\underline{X}}(\underline{x}) \log_2\left(\frac{1}{|a|^k} f_{\underline{X}}()\right) |a|^k d\underline{x}, \quad \text{letting } \underline{x} = \frac{\underline{y}-\underline{b}}{a}$$

$$= -\frac{1}{k}\int f(\underline{x}) \log_2 f(\underline{x})\, d\underline{x} - \frac{1}{k}\int f(\underline{x}) \log_2 |a|^k\, d\underline{x}$$

$$= h_{X,mk} + \log_2 |a|$$

$$\sigma_Y^2 = a^2\sigma_X^2$$

$$\Rightarrow \eta_{Y,k} = \frac{1}{\sigma_Y^2} 2^{2h_{Y,k}} = \frac{1}{a^2\sigma_X^2} 2^{2h_{X,k}+2\log_2|a|} = \frac{1}{\sigma_X^2} 2^{2h_{X,k}} = \eta_{X,k}$$

Notes:

- This property is consistent with the fact that scaling and shifting cause the opta to be scaled by $|a|^2$ and that the opta is proportional to 2^{2h}.
- We see that when $|a|$ is small, $h_{Y,k}$ is negative.

(3) (a) If $\underline{Y} = A\underline{X} + \underline{b}$ and A is a $k \times k$ nonsingular matrix, then

$$h_{Y,k} = h_{X,k} + \frac{1}{k}\log_2|A| \quad \text{and} \quad \eta_{Y,k} = \eta_{X,k}\,|A|^{2/k}\frac{\sigma_X^2}{\sigma_Y^2}$$

(b) If A is orthogonal (i.e. $A^{-1} = A^t$), then

$$h_{Y,k} = h_{X,k} \quad \text{and} \quad \eta_{Y,k} = \eta_{X,k}.$$

Derivation:

(a) Since $f_{\underline{Y}}(\underline{y}) = |A|^{-1} f_{\underline{X}}(A^{-1}(\underline{y}-\underline{b}))$,

$$h_{Y,k} = -\frac{1}{k}\int f(\underline{y}) \log_2 f(\underline{y})\, d\underline{y}$$

$$= -\frac{1}{k}\int |A|^{-1} f_{\underline{X}}(A^{-1}(\underline{y}-\underline{b})) \log_2\left(|A|^{-1} f_{\underline{X}}(A^{-1}(\underline{y}-\underline{b}))\right) d\underline{y}$$

$$= -\frac{1}{k}\int |A|^{-1} f_{\underline{X}}(\underline{x}) \log_2\left(|A|^{-1} f_{\underline{X}}(\underline{x})\right) |A|\, d\underline{x}, \quad \text{with } \underline{x} = A^{-1}(\underline{y}-\underline{b})$$

$$= -\frac{1}{k}\int f(\underline{x}) \log_2 f(\underline{x})\, d\underline{x} - \frac{1}{k}\int f(\underline{x}) \log_2 |A|^{-1} d\underline{x} \quad = \quad h_{X,k} + \frac{1}{k}\log_2|A|$$

$$\Rightarrow \quad \eta_{Y,k} = \frac{1}{\sigma_Y^2} 2^{2h_{Y,k}} = \frac{1}{a^2\sigma_X^2} 2^{2h_{X,k}+\frac{1}{k}\log_2|A|} = \frac{1}{\sigma_X^2} 2^{2h_{X,k}} |A|^{2/k}\frac{\sigma_X^2}{\sigma_Y^2} = \eta_{X,k}\,|A|^{2/k}\frac{\sigma_X^2}{\sigma_Y^2}$$

(b) This follows from (a) and the facts that when A is orthogonal, $|A| = 1$ and $\sigma_Y^2 = \sigma_X^2$.

(4) Gaussian: If $\underline{X} = (X_1,\ldots,X_k)$ is Gaussian with covariance matrix K, then

$$h_k = \frac{1}{2}\log_2 2\pi e|K|^{1/k} \quad \text{and} \quad \eta_k = 2\pi e\frac{|K|^{1/k}}{\sigma^2}$$

Notes:

- This formula for η_k is the same as the formula for β_k except that $((k+2)/k)^{(k+2)/2}$ is replaced by "e". (Note: $((k+2)/k)^{(k+2)/2} \to e$ as $k \to \infty$.)
- $|K|^{1/k}$ is nonincreasing in k, as will be shown later in the transform coding or dpcm sections of the notes.

Derivation:

We may assume $\underline{X}$ has zero mean, since previous properties show the mean has no effect on h_k or η_k. Let $\underline{Y} = A\underline{X}$, where A is the Karhunen-Loeve Transform.

$\underline{Y}$ is Gaussian with covariance matrix $\Lambda = AKA^t$, which is diagonal, with diagonal elements equal to the eigenvalues $\lambda_1,\ldots,\lambda_k$ of K.
Thus $\underline{Y}$ has independent components with variances $\lambda_1,\ldots,\lambda_k$.

Note that $|K| = \prod_{i=1}^{k}\lambda_i$.

$$h_{X,k} = h_{Y,k} = -\frac{1}{k}\int f(\underline{y}) \log_2 f(\underline{y})\, d\underline{y}$$

$$= -\frac{1}{k}\int f(\underline{y}) \log_2\left(\left(\prod_{i=1}^{k} 2\pi\lambda_i\right)^{-1/2} \exp\left\{-\sum_{i=1}^{k}\frac{y_i^2}{2\lambda_i}\right\}\right) d\underline{y}$$

$$= -\frac{1}{k}\log_2\left(\prod_{i=1}^{k} 2\pi\lambda_i\right)^{-1/2} + \frac{1}{k}\int f(\underline{y}) \sum_{i=1}^{k}\frac{y_i^2}{2\lambda_i} d\underline{y} \log_2 e$$

$$= -\frac{1}{k}\log_2 (2\pi)^{-k/2}|K| + \frac{1}{k}\sum_{i=1}^{k}\frac{EY_i^2}{2\lambda_i}\log_2 e$$

$$= \frac{1}{2}\log_2 2\pi|K|^{1/k} + \frac{1}{2}\log_2 e = \frac{1}{2}\log_2 2\pi e|K|^{1/k}$$

(5) If $\underline{X}$ has covariance matrix K (and is not necessarilty Gaussian), then

$$h_k \le \tfrac{1}{2}\log_2 2\pi e|K|^{1/k} \quad \text{and} \quad \eta_k \le 2\pi e\frac{|K|^{1/k}}{\sigma^2}$$

with equality iff $\underline{X}$ is Gaussian

Derivation:

We may assume $\underline{X}$ has zero mean, since mean has no effect on h_k or η_k.

Let $f(\underline{x})$ be the density of $\underline{X}$ with covariance matrix K, and let $g(\underline{x})$ be the Gaussian density with mean zero and covariance matrix K.

We make the proof in two steps.

(a) $h_k(f) \le -\frac{1}{k}\int f(\underline{x}) \log_2 g(\underline{x})\, d\underline{x}$:

$$-\frac{1}{k}\int f(\underline{x}) \log_2 g(\underline{x})\, d\underline{x} - h_k(f)$$

$$= -\frac{1}{k}\int f(\underline{x}) \log_2\frac{g(\underline{x})}{f(\underline{x})}\, d\underline{x}$$

$$\ge -\frac{1}{k}\int f(\underline{x})\left(\frac{g(\underline{x})}{f(\underline{x})} - 1\right) d\underline{x}\,\frac{1}{\ln 2}, \qquad \left(\log_2 z \le (z-1)\frac{1}{\ln 2}\right)$$

$$= -\frac{1}{k}\int g(\underline{x})\, d\underline{x}\,\frac{1}{\ln 2} + \int f(\underline{x})\, d\underline{x}\,\frac{1}{\ln 2} = -1 + 1 = 0$$

(b) $-\frac{1}{k}\int f(\underline{x}) \log_2 g(\underline{x})\, d\underline{x} = -\frac{1}{k}\int g(\underline{x}) \log_2 g(\underline{x})\, d\underline{x} = \frac{1}{2}\log_2 2\pi e|K|^{1/k}$:

$$-\frac{1}{k}\int f(\underline{x}) \log_2 g(\underline{x})\, d\underline{x} = -\frac{1}{k} E \log_2\left((2\pi)^{-k/2}|K|^{-1/2}\exp\left\{-\frac{1}{2}\underline{X}^t K^{-1}\underline{X}\right\}\right) d\underline{x}$$

$$= \frac{1}{2}\log_2 2\pi|K|^{1/k} + \frac{1}{2k} E_f\left[\underline{X}^t K^{-1}\underline{X}\right] \log_2 e$$

where E_f denotes expecation with respect to f.

Since the expectation is a sum of terms of the form $a_{ij} E[X_i X_j]$, it depends only on the covariance matrix K. Therefore, it will be the same if the expectation is taken with respect to g, because g has the same covariance matrix. Therefore

$$-\frac{1}{k}\int f(\underline{x}) \log_2 g(\underline{x})\, d\underline{x} = \frac{1}{2}\log_2 2\pi|K|^{1/k} + \frac{1}{2k} E_g\left[\underline{X}^t K^{-1}\underline{X}\right] \log_2 e$$

$$= -\frac{1}{k}\int g(\underline{x}) \log_2 g(\underline{x})\, d\underline{x}$$

$$= \frac{1}{2}\log_2 2\pi e|K|^{1/k} \quad \text{by (4)}$$

Definition:

The *conditional differential entropy* of random variables $X_1,\ldots,X_k$ given random variables $Y_1,\ldots,Y_m$

$$h(X_1,\ldots,X_k|Y_1,\ldots,Y_m) \triangleq -\int f(\underline{x},\underline{y}) \log_2 f(\underline{x}|\underline{y})\, d\underline{x}\, d\underline{y}$$

Most of the following properties are derived in the same way as the corresponding propety for entropy.

(6) $h(X|Y) \le h(X)$ with equality iff X and Y are independent.

Derivation: We'll show $h(X) - h(X|Y) \ge 0$ with equality iff X indep of Y.

$$h(X) - h(X|Y) = -\int f(x,y) \log_2 f(x)\, dx\, dy + \int f(x,y) \log_2 \frac{f(x,y)}{f(y)}\, dx\, dy$$

$$= -\int f(x,y) \ln \frac{f(x)f(y)}{f(x,y)}\, dx\, dy\, \frac{1}{\ln 2}$$

$$\ge -\int f(x,y) \left(\frac{f(x)f(y)}{f(x,y)} - 1\right) dx\, dy\, \frac{1}{\ln 2} \quad \text{since } \ln z \le z-1$$

$$= -\int f(x)f(y)\, dx\, dy\, \frac{1}{\ln 2} + \int f(x,y)\, dx\, dy\, \frac{1}{\ln 2} = 0.$$

Equality holds if and only if $f(x)f(y) = f(x,y)$ for all x,y; i.e. if and only if X and Y are independent.

(7) $h(Y_1,\ldots,Y_n|X_1,\ldots,X_m) \le h(Y_1,\ldots,Y_n|X_1,\ldots,X_{m'})$, $0 \le m' < m$,

with equality iff $Y_1,\ldots,Y_n$ is conditionally independent of $X_{m'+1},\ldots,X_m$ given $X_1,\ldots,X_{m'}$.

Derivation: Similar to that of (6).

(8) Chain rule:

$$h(X_1,\ldots,X_k) = h(X_1)+h(X_2|X_1)+H(X_3|X_1X_2)+ \ldots +h(X_k|X_1\ldots X_{k-1})$$

Derivation: Essentially the same proof as for the chain rule for ordinary entropy, but with H's replace by h's.

(9) $h(X_1,\ldots,X_k) \le h(X_1)+ \ldots + h(X_k)$

with equality if and only if X_i's are independent

Derivation: Essentially the same proof as for the analogous property for ordinary entropy.

STATIONARY SOURCES

Definitions:

$$h_k = \frac{1}{k} h(X_1,\ldots,X_k)$$

$$h_{1|m} = h(X_n|X_{n-m},X_2,\ldots,X_{n-1}) \qquad (\underline{h}_1 = h(X_1) = h_1)$$

$$h_{k|m} = \frac{1}{k} h(X_n^{n+k-1}|X_{n-m}^{n-1}) \qquad (h_{1|k} = \underline{h}_1)$$

$$h_\infty = \lim_{k\to\infty} h_k = \text{differential entropy-rate of } X$$

Properties:

(10) $h_{1|k+1} \le h_{1|k}$

Derivation: Follows from (7) and stationarity.

(11) $h_k = \frac{1}{k}(h_1+h_{1|1}+h_{1|2}+\dots+h_{1|k-1}) \ge h_{1|k-1} \ge h_{1|k}$

Derivation: Essentially the same as the analogous property for entropy.

$$h_k = \frac{1}{k} h(X_1,\dots,X_k)$$
$$= \frac{1}{k}\Big(h(X_1)+h(X_2|X_1)+h(X_3|X_1,X_2) + \dots + h(X_k|X_1,X_2,\dots,X_{k-1})\Big) \quad \text{chain rule (8)}$$
$$= \frac{1}{k}\Big(h_1+h_{1|1}+h_{1|2}+\dots+h_{1|k-1}\Big) \quad \text{by stationarity}$$
$$\ge \frac{1}{k}\Big(h_{1|k-1}+h_{1|k-1}+h_{1|k-1}+\dots+h_{1|k-1}\Big) = h_{1|k-1} \ge h_{1|k} \quad \text{by (10)}$$

(12) $h_{k+1} \le h_k$

It follows that $h_\infty = \lim_{k\to\infty} h_k$ is a well-defined quantity, because the h_k's are nonincreasing, they must have a limit. (It could be a finite value or $-\infty$.)

Derivation: By (11), h_k is the average of the k terms $h_1, h_{1|1}, h_{1|2},\dots,h_{1|k-1}$. Similarly, h_{k+1} is the average of the $k+1$ terms $h_1, h_{1|1}, h_{1|2},\dots,h_{1|k-1}, h_{1|k}$. Since the extra term in h_{k+1} is no larger than all other terms, $h_{k+1} \le h_k$.

(13) $\lim_{k\to\infty} h_{1|k} = h_\infty$

Derivation: Since the $\underline{h}_k$'s are nonincreasing with k, they have a limit. (It could be a finite value or $-\infty$.)

Since by Prop. 10, h_k is the average of the k terms $h_1, h_{1|1}, h_{1|2},\dots,h_{1|k-1}$, the limit of the $h_{1|k}$'s equals the limit of the h_k's, which by definition is h_∞.

(14) For an IID source,

$$h_1 = h_k = h_\infty \quad \text{all } k$$

Derivation: This follows from (9).

(15) For a stationary Markov source

$$h_k = \frac{1}{k} h(X_1) + \frac{k-1}{k} h(X_2|X_1)$$

$$h_\infty = h(X_2|X_1) = h_{1|1}$$

Derivation: See the proof of (11).

(16) For a stationary, first-order, autoregressive source, $\left(X_k = \rho X_{k-1} + W_k\right.$ where the W_k's are IID and independent of past X_k's$\left.\right)$, (it's Markov, too),

$$h_k = \frac{1}{k} h(X_1) + \frac{k-1}{k} h(W_1)$$

$$h_\infty = h(X_2|X_1) = h(W_1) \quad \text{and} \quad \eta_\infty = \eta_{W,1}.$$

Derivation: Use (15) and the fact that

$$h(X_2|X_1) = h(aX_1+W_2|X_1) = h(W_2|X_1) \quad \text{(because } aX_1 \text{ is a constant)}$$

$= h(W_2)$ because W_2 and X_1 are independent

$= h(W_1)$ because W_2 and W_1 are independent

(17) For a stationary Gaussian source with power spectral density $S(\omega)$,

$$h_\infty = \frac{1}{2} \log_2 2\pi e Q \quad \text{and} \quad \eta_\infty = 2\pi e \frac{Q}{\sigma^2}$$

where $Q = \exp\left\{\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln S(\omega)\, d\omega\right\}$ = one-step prediction error

Derivation: From (4)

$$h_\infty = \lim_{k\to\infty} h_k = \lim_{k\to\infty} \frac{1}{2} \log_2 2\pi e |K|^{1/k} = \frac{1}{2} \log_2 2\pi e Q$$

where $\lim_{k\to\infty} |K|^{1/k} = Q$ will be demonstrated later in the context of transform coding.

(18) For any stationary source

$$h_\infty \le \frac{1}{2} \log_2 2\pi e Q \quad \text{and} \quad \eta_\infty \le 2\pi e \frac{Q}{\sigma^2}$$

with equality if only if the source is Gaussian.

Derivation: This follows from $h_k \le h_{k,\text{Gauss}}$ for all k (see (5)) and the fact that

$$h_\infty = \lim_{k\to\infty} h_k \le \lim_{k\to\infty} h_{k,\text{Gauss}} = h_{\infty,\text{Gauss}} = \frac{1}{2} \log_2 2\pi e Q.$$

(19) For a stationary, first-order autoregressive, Gaussian source with correlation coefficient ρ

$$h_k = \frac{1}{2}\log_2 2\pi e\sigma^2 + \frac{k-1}{k}\frac{1}{2}\log_2(1-\rho^2)$$

$$h_\infty = \frac{1}{2}\log_2 2\pi e\sigma^2(1-\rho^2) \quad \text{and} \quad \eta_\infty = 2\pi e(1-\rho^2)$$

Derivation: Uses (16).

(20) For a stationary source, $\eta_\infty = \beta_\infty$

Sketch of Derivation: Uses following fundamental result of information theory:

Asymptotic Equipartition Property (AEP):

For a stationary ergodic source and all sufficiently large k,

$$\Pr(\underline{X} \in T_k) \cong 1$$

where $T_k = \{\underline{x} : \frac{1}{k}\log_2 f_k(\underline{x}) \cong h_\infty\} = \{\underline{x} : f_k(\underline{x}) \cong 2^{-kh_\infty}\}$

That is, $f_k(\underline{x}) \cong 2^{-kh_\infty}$ with high probability.

Here's how we use it:

$$\log_2 \sigma^2\beta_\infty \cong \log_2 \sigma^2\beta_k = \log_2\left(\int f_k(\underline{x})^{k/(k+2)}d\underline{x}\right)^{(k+2)/k} \quad \text{for large values of } k$$

$$= \frac{k+2}{k}\log_2 \int f_k(\underline{x})^{-2/(k+2)} f_k(\underline{x})\, d\underline{x} \cong \frac{k+2}{k}\log_2 \int_{T_k} f_k(\underline{x})^{-2/(k+2)} f_k(\underline{x})\, d\underline{x}$$

because $\Pr(\underline{X} \in T_k) \cong 1$

$$\cong \frac{k+2}{k}\log_2 \left(2^{-kh_\infty}\right)^{-2/(k+2)}$$ because $f_k(\underline{x}) \cong 2^{-kh_\infty}$ for $\underline{x} \in T_k$.

$$\cong 2h_\infty$$ since k is large.

$$\Rightarrow \quad \beta_\infty = \frac{2^{2h_\infty}}{\sigma^2} = \eta_\infty$$

Why Does VL-VQ Attain the Performance That It Does?

Compare the point density and cell shapes of

$Q_{k'}$ = optimal k'-dim'l VL-VQ

Q_k = optimal k-dim'l -VL-VQ, with k = large multiple of k'.

$Q_{k'}$ has constant inertial profile $m^*_{k'}$, with uniform pt. density & inert'l profile.

Q_k has constant inertial profile $m^*_k \cong 1/(2\pi e)$.

To compare them, consider the k-dimensional product quantizer $Q_{pr,k}$ that is formed by using $Q_{k'}$ k/k' times in succession.

$Q_{pr,k}$ has the same rate, distortion and "shortcomings" as $Q_{k'}$.

$\lambda_{pr,k}(\underline{x})$ = uniform point density in dimension k = best point density

$m_{pr,k}(\underline{x}) = m^*_{k'} > m^*_k$

We see that the only shortcoming of k'-dimensional VL-VQ relative to high-dimensional VL-VQ is the space-filling loss:

$$L_{sp} = \frac{m^*_{k'}}{1/(2\pi e)} .$$

With VL-VQ, there is no point density loss, even for sources with memory. In effect, entropy coding has eliminated the need to compromise between good point density and small oblongitis.

Why VL-VQ and FL-VQ Give Same Performance for Large k?

Consider their properties on $T_k = \{\underline{x} : f_k(\underline{x}) \cong 2^{-kh_\infty}\}$, which has $\Pr(\underline{X} \in T_k) \cong 1$

Cell shapes: Both have $m(\underline{x}) \cong \frac{1}{2\pi e}$ = NMI of high dim'l sphere

Point density:

VL-VQ: $\lambda(\underline{x})$ = constant

FL-VQ: $\lambda^*_k(\underline{x}) = c\, f_k^{k/(k+2)}(\underline{x}) \cong f_k(\underline{x}) \cong \begin{cases} 2^{-kh_\infty}, & \underline{x} \in T_k \\ 0, & \text{else} \end{cases}$

$\Rightarrow$ VL-VQ and FL-VQ have same inertial profile and same point density in T_k, where it matters.

Binary encoders

FL-VQ: all codewords have length kR

VL-VQ: codeword for cell $S_{\underline{x}}$ containing $\underline{x} \in T_k$ has

length $\cong -\log_2 \Pr(S_{\underline{x}}) \cong -\log_2 \left(f_k(\underline{x}) \times |S_{\underline{x}}|\right) \cong kh_\infty - \log_2 \frac{1}{\Lambda(\underline{x})}$ for $\underline{x} \in T_k$

$\cong$ constant $\cong kR$

$\Rightarrow$ VL-VQ and FL-VQ assign same lengths to $\underline{x}$ in T_k

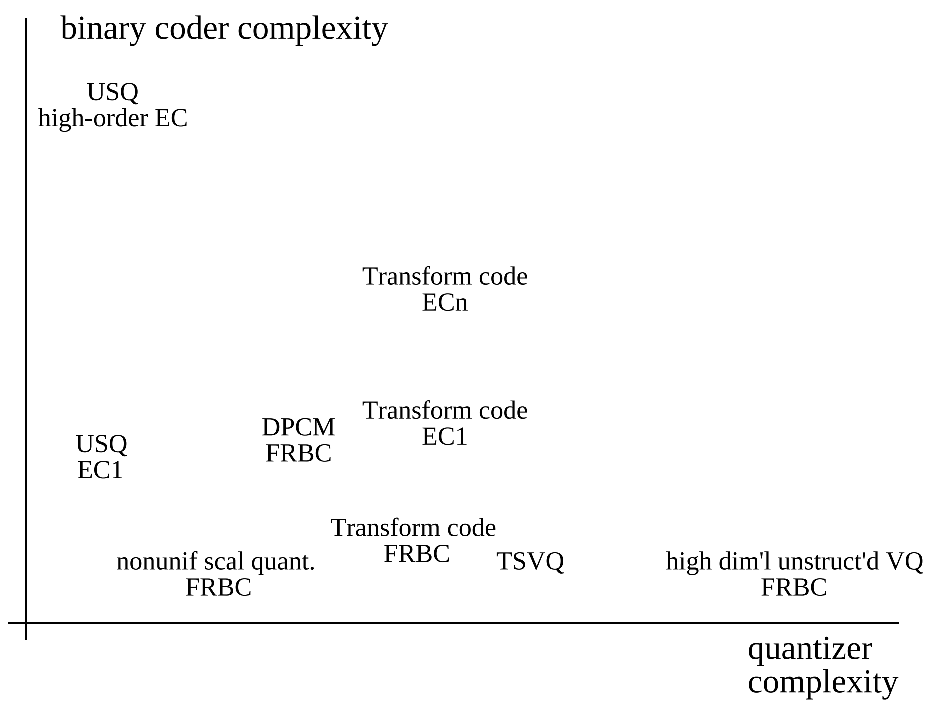
- FL-VQ "uses" pt. density to "ignore" $\underline{x}$'s not in T_k.
- VL-VQ assigns short codewords to T_k.

Second view:

FL-VQ: $\Pr(\text{cell containing } \underline{x}) = \Pr(S_{\underline{x}}) \cong f_k(\underline{x})\,|S_{\underline{x}}| \cong f_k(\underline{x}) \frac{1}{M\lambda_k^*(\underline{x})} \cong \frac{1}{M}$

Since all cells have the same probability, variable-length coding gains nothing.

Qualitative plot of quantizer and entropy coder complexity



FRBC = Fixed-Rate Binary Code

ECn = nth-order Entropy Code

Positions of codes are subjective. Please don't quote me.