Vector Quantization-

→ In vector quantization, we group the source output into blocks or vectors.

→ This vector of source output is given to vector quantizer as input.

→ If L source outputs are combined together, wheather it be audio (or speech) or image, then it forms a L-dimensional vector.

→ At both encoder and decoder of vector quantizer, we have a set of L-dimensional vectors called the codebook of vector quantizer.

→ Each vector in vector codebook, is known as code vector.

→ Each code vector is assigned a binary index. At the encoder, the input vector is compared to each code-vector in-order to find out the code vector closest to input vector.

→ To inform the decoder about which code-vector was selected by the encoder, we transmit the binary index of the code-vector.

→ If no. of codevectors (i.e size of codebook) is K, then number of bits per vector = $\lceil \log_2 K \rceil$ bits.

→ Rate of L-dimensional vector quantizer with a codebook of size K is $\dfrac{\lceil \log_2 K \rceil}{L}$.

→ Decoder uses a look-up table to find the code-vector corresponding to binary index.

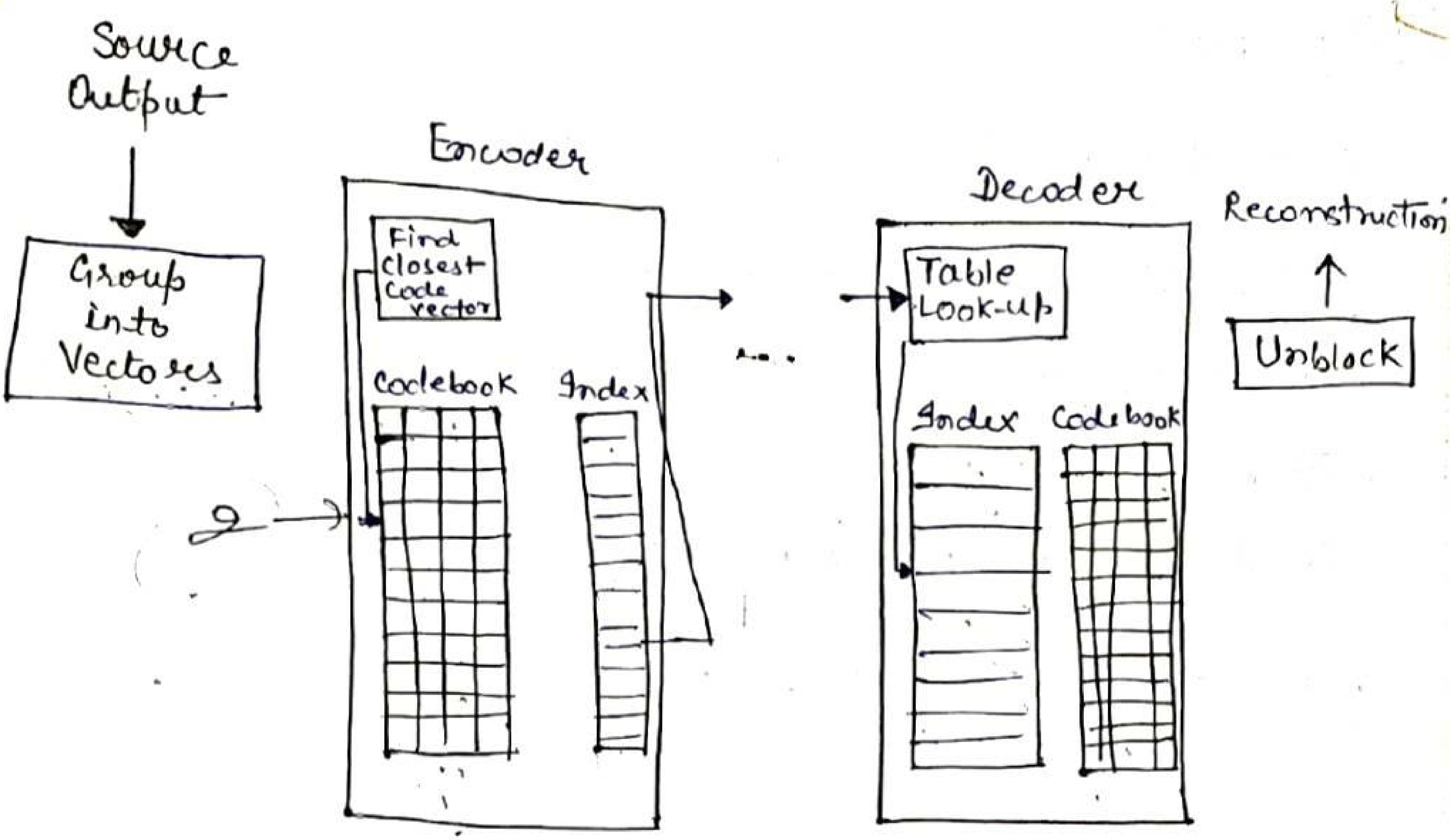


Figure: Vector quantization procedure

Measure of distortion →

As a measure of distortion, we use the mean squared value. If in a codebook E, containing k code vectors $\{Y_i\}$, the input vector X is closest to $\{Y_j\}$, then it means that –

$$|X - Y_j|^2 \le |X - Y_i|^2 \quad \text{for all } Y_i \in E$$

where $X = \{x_1, x_2, x_3, \cdots, x_L\}$

and

$$\boxed{|X|^2 = \sum_{i=1}^{L} x_i^2}$$

Advantages of Vector Quantization Over Scalar Quantization:-

(1) In scalar quantization, each input symbol is treated separately in producing the output, while in vector quantization, the input symbols are clubbed together in groups called vectors and processed to give the output.

(2) For same rate, use of vector quantization results in lower distortion than scalar quantition.

(3) In scalar quantization, decision boundary is denoted by $\{b_i\}_{i=0}^{M}$ and reconstruction level by $\{y_i\}_{i=1}^{M}$, and quantization operation by $Q(\cdot)$. Then,

$$Q(x) = y_i \quad \text{iff} \quad b_{i-1} < x \leq b_i$$

Mean square quantization error is given by-

$$\sigma_q^2 = \int_{-\infty}^{\infty} (x - Q(x))^2 f_x(x)\, dx$$

$$= \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} (x - y_i)^2 f_x(x)\, dx$$

In vector quantization, quantization process is shown as-

$$Q(x) = Y_j \quad \text{iff} \quad d(x, Y_j) < d(X, Y_i) \ \forall \ i \neq j$$

Quantization region is defined as-

$$V_j = \{X : d(X, Y_j) < d(X, Y_i) \ \forall \ i \neq j\}$$

Initial Set of Output points -

Height	Weight
45	50
75	117
45	117
80	180

(3) Compute the average distortion $D^{(K)}$ between the training vectors and representative reconstruction value.

$$D^{(K)} = \sum_{i=1}^{M} \int_{V_i^{(K)}} |x - Y_i^{(K)}|^2 f_x(x)\,dx$$

(4) If $\frac{D^{(K)} + D^{(K-1)}}{D^{(K)}} < \varepsilon$, stop; otherwise continue.

(5) $K = K+1$. Find new reconstruction values $\{Y_i^{(K)}\}_{i=1}^{M}$ that are the average value of the elements of each of the quantization regions $V_i^{(K-1)}$. Goto step 2.

→ Also, known as generalized Lloyd algorithm.

Initializing LBG algorithm –

→ LBG algorithm gurantees that distortion from one iteration to the next will not increase.

Ex- Suppose training set consists of height and weight set of output points as shown in as shown in table "1", and input-output table "2",

Height	Weight
72	180
65	120
59	119
64	150
65	162
57	88
72	175
44	41
62	114
60	110
56	91
70	172

(4) In scalar quantization, the granular error was determined by the size of quantization interval.
In vector quantization, the granular error is affected by the size and shape of quantization interval.

The LINDE-BUZO-GRAY algorithm—

→ LBG algorithm is popular approach for obtaining vector quantizer codebook.

→ Most popular approach to design vector quantizer is a clustering procedure known as K-means algorithm,

→ LBG algorithm is quite similar to K-means algorithm.

→ LBG algorithm works as follows—

(1) Start with an initial set of reconstruction values $\{Y_i^{(0)}\}_{i=1}^{M}$, and a set of training vectors $\{X_n\}_{n=1}^{N}$.
Set $K=0$, $D^{(0)}=0$. Select threshold ϵ.
distortion

(2) The quantization regions $\{V_i^{(K)}\}_{i=1}^{M}$ are given by -

$$V_i^{K} = \{X_n : d(X_n, Y_i) < d(X_n, Y_j) \ \forall \ j \neq i\}$$
$$i = 1, 2, \ldots M.$$

Tree-Structured vector quantizer

(i) For Symmetrical Situations →

→ Structure means how we organise our codebook. It should be designed such that it is easy to pick which part contains the desired output vector.

→ Consider the following vector quantizer –

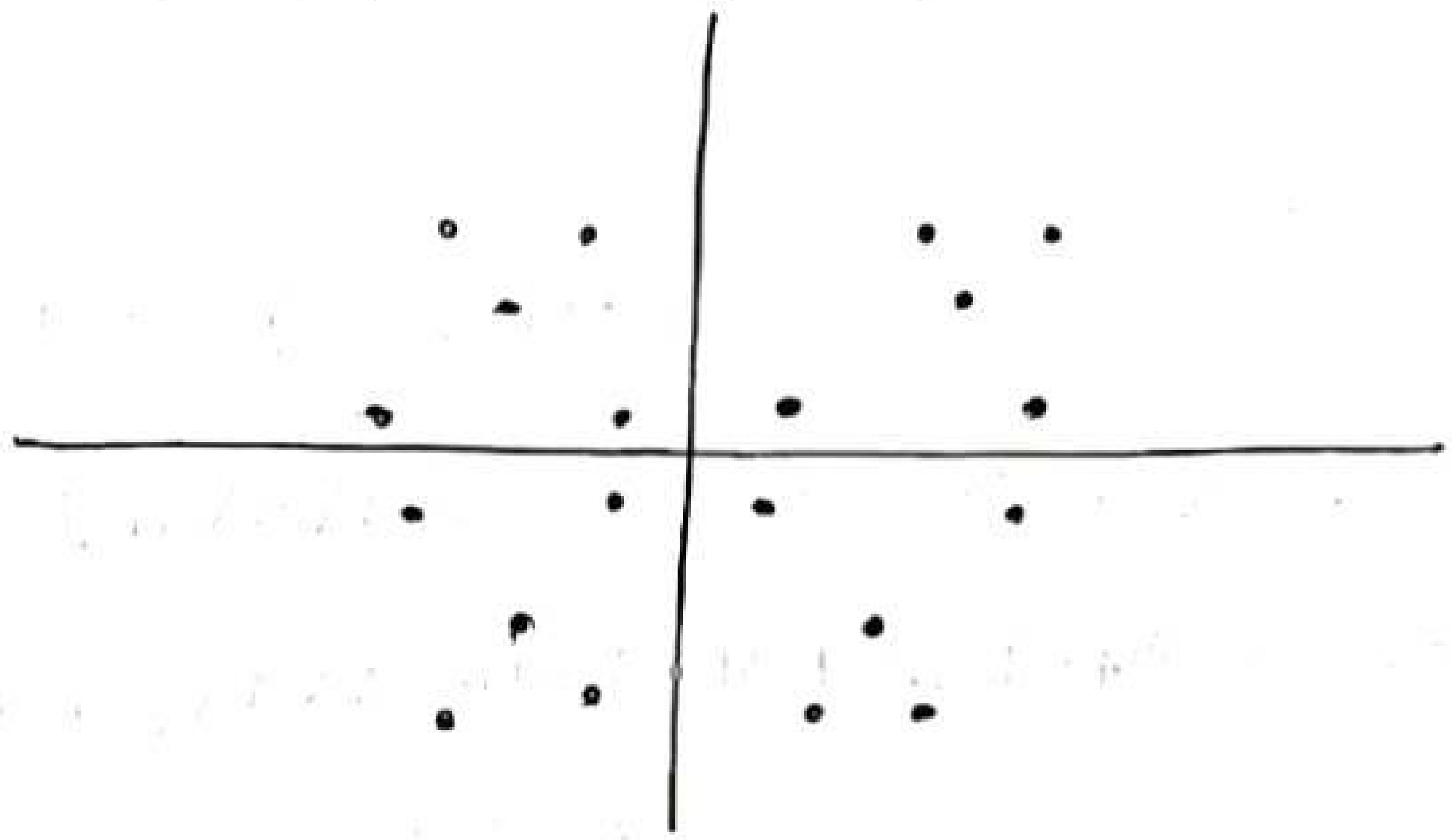


fig:- A Symmetrical vector quantizer in 2-D

→ Output points in each quadrant are the mirror images of the output points in neighbouring quadrant.

→ If an input vector is given to this quantizer, no. of comparisions to find the closest output point by using the sign on the components of input.

→ The sign on the components of input vector will tell us in which quadrant the input lies.

→ Because all the quadrants are mirror images of the neighbouring quadrants, ~~the closest output point to a given input will~~ therefore, we only need to compare input to output points that lie in the same quadrant, thus reducing the no. of comparisions by a fac[redacted]4.

(ii) Tree - Structural for Non- Symmetrical Situations →

→ Divide the set of output points into two groups - group 0 and group 1.

→ Assign each group a test vector such that output points in each group are closer to test vector assigned to that group.

→ When we get an input vector, we compare it against the test vectors. Depending upon the outcome, the input is compared to output points.

→ Test vector and input vector is compared by looking at the sign of components.

→ If total no. of output points is K, with this approach, we have to make $\frac{K}{2}+2$ comparisions instead of K comparisions.

→ This process can be continued by spliting the output points in each group and assigning a test vector to the sub-groups. So, group 0 will be split into group 00 and group 01, and so on.

→ Process is continued until last set of group would consist of single points (if O/P points are in power of 2)

→ Thus no. of comparisions required to obtain final output point would be $\underline{2 \log_2 K}$ instead of K.

→ For a codebook of size 4096, we need,

$$= 2 \log_2 (4096)$$
$$= 2 \times \log_2 (2^{12})$$
$$= 2 \times 12 = \underline{24} \text{ vector comparisions.}$$

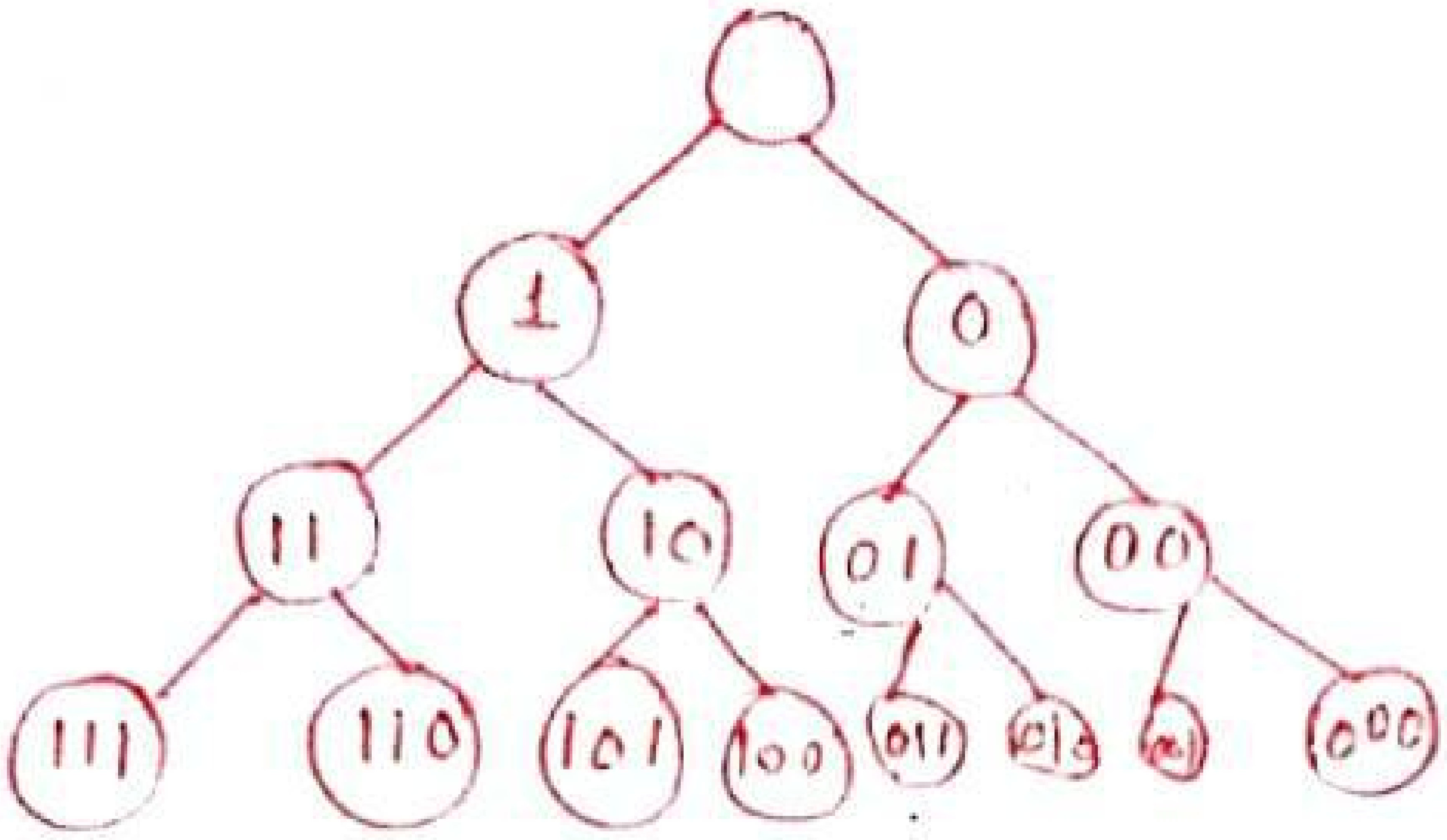


Fig:- Decision tree for quantization

(iii) Pruned Tree Structured Vector Quantizer →

→ Once we have built a tree-structured codebook, we can improve its rate distortion performance by removing carefully selected sub-groups.

→ Removal of sub-groups is referred to as pruning.

→ Prunning will reduce the size of the codebook and hence the rate.

→ The objective of pruning is to remove those sub-groups that will result in the best trade-off rate & distortion.

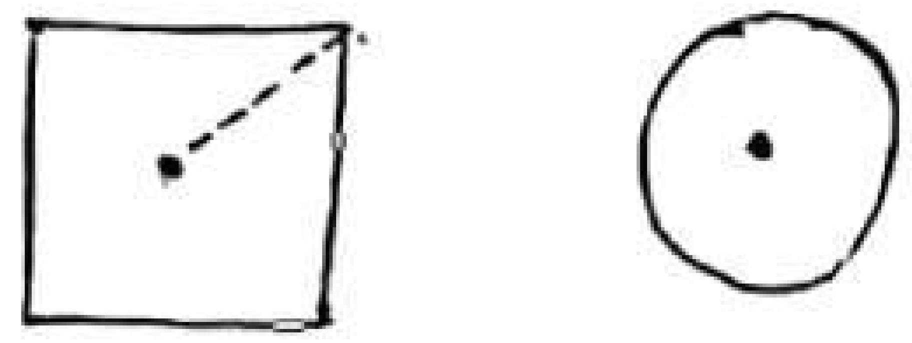
→ Pruning results in variable-length codewords.

→ These variable length codes corresponds to the leaves of a binary tree.

Lattice Vector Quantizer–

→ In vector quantization, the granular error is affected by the size and shape of the quantization interval.

→ Consider the following square and circular quantization regions–



→ For simplicity, we consider the quantization region only at the origin.

→ Assume, that both the shapes have same area. So, quantization region to cover this area will also be the same.

→ If area $= 1$,
in case of circle, radius $= \frac{1}{\sqrt{\pi}}$,

$$\text{Side of Square} = 1.$$

→ In case of square, maximum possible quantization error $= \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = \frac{1}{\sqrt{2}}$

$$\simeq 0.707$$

→ In case of circular region, maximum possible quantization error $= \frac{1}{\sqrt{\pi}} \simeq 0.56$

→ If we compute, the average squared error for the square region, we obtain–

$$\int_{\text{square}} \|x\|^2 \, dx = 0.1666$$

where C is a constant depending on the variance of the input.

Polar and Spherical Vector Quantizers →

→ For the Gaussian distribution, the contours of constant probability are circles in two dimensions and spheres and hyperspheres in three and higher dimensions.

→ In 2-D, we can quantize the input vector by first transforming it into polar co-ordinates r and θ.

$$r = \sqrt{x_1^2 + x_2^2}$$

and
$$\theta = \tan^{-1}\frac{x_2}{x_1}$$

Then r and θ can be quantized independently, or we can use the quantized value of r as an index to a quantizer for θ. The former is known as polar quantizer and latter an unrestricted polar quantizer.

Structured Vector Quantizer

Structured vector quantizer can be of three types—
(According to shape)

(i) Pyramid vector quantizer

(ii) Polar & spherical vector quantizer

(iii) Lattice vector quantizer

Pyramid Vector quantizer—

→ Suppose, we are quantizing a random variable X with pdf $f_X(x)$ and differential entropy $h(X)$, and suppose vector corresponding to random variable is vector $\mathbf{X}$.

→ According to Shannon's Asymptotic Equipartition property (AEP), states that for sufficiently large L and arbitrarily small ϵ,

$$\boxed{\left|\frac{\log f_X(X)}{L} + h(X)\right| < \epsilon}$$

→ When almost all L-dimensional vectors will lie on a contour of constant probability given by —

$$\boxed{\left|\frac{\log f_X(X)}{L}\right| = -h(X)}$$

The vector quantizer consists of points of the rectangular quantizer that fall on the hyperpyramid given by —

$$\boxed{\sum_{i=1}^{L} |x_i| = C}$$

Scanned by TapScanner