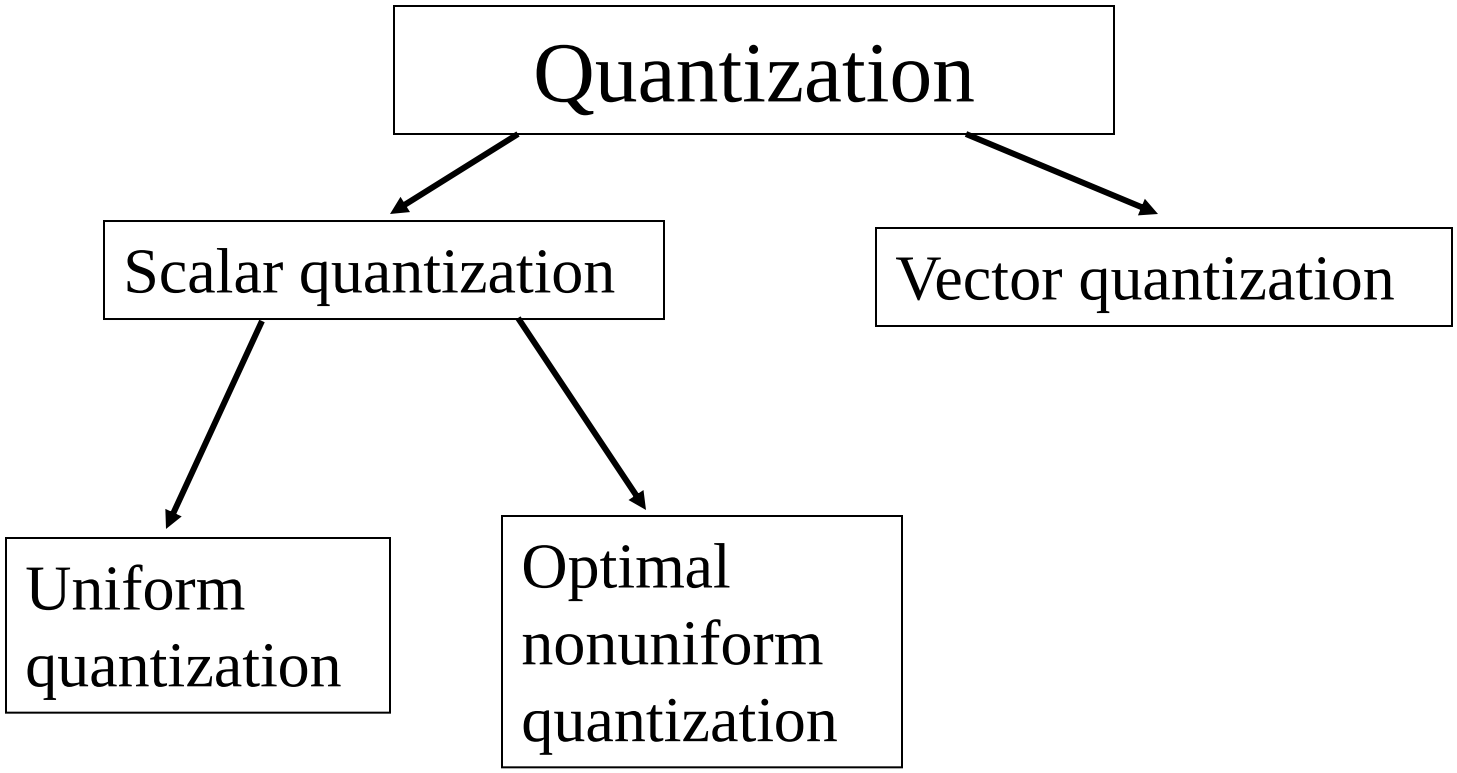


The second of the two operations required to change an analog signal into the digital signal is called quantization. We can say that quantization is a procedure of transforming sample values into integer values with evidently finite length of binary representation.

Scalar quantization

Scalar quantization is a mapping Q of real value x of a continuous random variable X into the closest (in terms of the chosen distortion measure $d(x, y)$) value $y = Q(x)$ from a discrete set $Y = \{y_1, ..., y_M\}$.

The values $\{y_i\}$ are called *output levels, reproduction levels, or approximating values.*

Y is called *approximating set or codebook.*

Scalar quantization

Scalar quantizer is determined by a set of *thresholds* $\{t_i\}, i = 0,1,...,M, t_i < t_{i+1}$ which split the real line R into subintervals or *cells* $\Delta_i = (t_{i-1}, t_i]$, such that:

$\Delta_i \bigcap \Delta_j = \varnothing$ and $\bigcup_{i=1}^{M} \Delta_i = R,$

and *approximating values* chosen in such a way that:

$y_i = Q(x)$ if and only if $x \in \Delta_i.$

Scalar quantization

The two outermost cells $(t_0, t_1]$ and (t_{M-1}, t_M) can be finite if we quantize an interval of R or infinite if R is quantized.

In the first case M is a finite number and a set of of cells is finite. In the second case a set of cells can be a countable set, that is, M can be infinite.

Scalar quantization

The general definition reduces to the rounding off example if $\Delta_i = (i - 1/2, i + 1/2]$ and $y_i = i$ for all integers i.

The quantization procedure:

- x is sequentially compared with $t_i, i = 1,2,...,M$

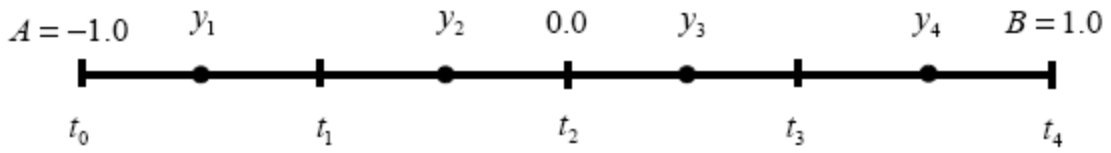
  to specify the cell Δ_i which contains x.

- The binary representation of i is stored or transmitted over a communication channel.

Example

$t_0 = -1.0 \quad t_1 = -0.5 \quad t_2 = 0.0 \quad t_3 = 0.5 \quad t_4 = 1.0$

$y_1 = -0.75 \quad y_2 = -0.25 \quad y_3 = 0.25 \quad y_4 = 0.75$



$R = 2$ bits.

$x = 0.285 \quad i = 2 \rightarrow 10 \rightarrow y_3 = 0.25$

$$d(x, y) = (x - y)^2 = 1.2 \times 10^{-3}$$

Rate-distortion function

The quality of any scalar quantizer is measured by a rate-distortion function $R(D)$, where

$$D(Q) = E\{d(x, y)\}$$

is the average quantization error,

$d(x, Q(x))$ is a fidelity criterion or distortion measure.

The most commonly used: $d(x, y) = (x - y)^2$.

Rate-distortion function

The quantization rate R of a scalar quantizer is the the number of bits required to represent the value x.

For fixed-rate quantizer: $R = \log_2 M$ bits/sample.

For variable-rate quantizer: $$R = -\sum_{i=1}^{M} P(y_i) \log_2 P(y_i),$$

where $P(y_i) = \int_{t_{i-1}}^{t_i} f(x)dx$ is the probability of the i th reproduction level, $f(x)$ is a pdf of X.

Uniform scalar quantization

A quantizer is said to be uniform if

$|\Delta_i| = t_i - t_{i-1} = \delta, \ \ i = 1,2,...,M,$ where

δ is quantization step, $y_i = \dfrac{t_{i-1} + t_i}{2}$

is in the middle of the subinterval.

If all M codewords have length $\log_2 M$ bits

we obtain fixed-rate quantizer.

Uniform scalar quantization

The most commonly used uniform variable-rate quantizer is based on rounding off:

$$i = \left[\frac{x}{\delta} \right]$$ is the cell number,

the approximating value corresponding to the input

x is computed as $y = Q(x) = \delta \left[\frac{x}{\delta} \right].$

$$\Delta_i = (i\delta - \delta/2, i\delta + \delta/2], y_i = i\delta.$$

Scalar quantization demo

Original file

Variable-rate quantizer compression ratio=2.14, relative error 4×10^{-5}

Variable-rate quantizer compression ratio=4.5, relative error 1.0×10^{-2}

Variable-rate quantizer compression ratio=6.4, relative error 4.1×10^{-2}

Scalar quantization demo

Original file

Variable-rate quantizer compression ratio=2.14, relative error 1.2×10^{-5}

Variable-rate quantizer compression ratio=4.5, relative error 3.2×10^{-3}

Variable-rate quantizer compression ratio=6.1, relative error 1.0×10^{-2}

Nonuniform scalar quantization

Consider fixed-rate scalar quantizer. If pdf $f(x)$ is known then MSE

$$D = \int_{-\infty}^{\infty} (x - Q(x))^2 f(x)dx = \sum_{i=1}^{M} \int_{t_{i-1}}^{t_i} (x - y_i)^2 f(x)dx.$$

For the given M MSE is a function of $2M-1$ parameters ($\{t_1,...,t_{M-1}\}$, $\{y_1,...,y_M\}$.)

We can rewrite the formula for MSE

$$D = \int_{-\infty}^{\infty} x^2 f(x)dx - 2\sum_{i=1}^{M} y_i \int_{t_{i-1}}^{t_i} xf(x)dx + \sum_{i=1}^{M} y_i^2 \int_{t_{i-1}}^{t_i} f(x)dx$$

Nonuniform scalar quantization

By differentiating with respect to $y_i's$ and $t_i's$ setting derivatives equal to 0 we obtain

$$-2\int_{t_{i-1}}^{t_i} xf(x)dx + 2y_i\int_{t_{i-1}}^{t_i} f(x)dx = 0$$

$$-2y_i t_i f(t_i) + y_i^2 f(t_i) + 2y_{i+1}t_i f(t_i) - y_{i+1}^2 f(t_i) = 0$$

The solution of optimization problem is

$$t_i = (y_i + y_{i+1})/2, \quad (2.1)$$

$$y_i = \frac{m_i}{P(y_i)} \quad (2.2)$$

$$m_i = \int_{t_{i-1}}^{t_i} xf(x)dx. \quad (2.3)$$

$$D_{opt} = E - \sum_{i=1}^{M} \frac{m_i^2}{P(y_i)} \quad (2.4)$$

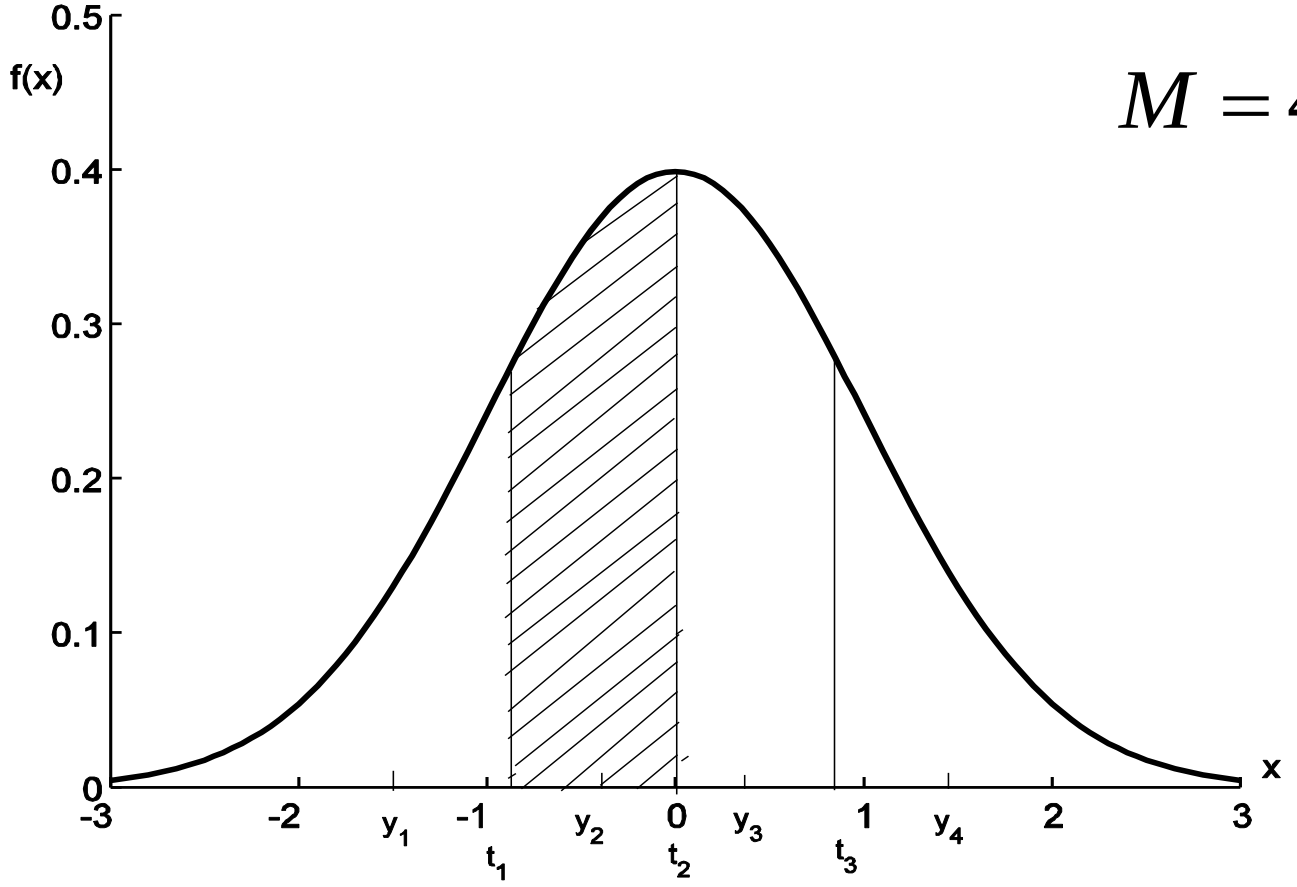
Nonuniform scalar quantization

Formula (2.2) determines the centroid of the area of the pdf $f(x)$ between t_{i-1} and t_i. The quantization procedure described by (2.1)-(2.4) is called nonuniform optimal scalar quantization. It was independently suggested by S. P. Lloyd and J. Max. J. Max also designed fixed-rate nonuniform optimal quantizer for the Gaussian random variable X, that is,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-(x-m)^2/(2\sigma^2)\},$$

where m is mean value and σ^2 is the variance of X.

Example



$M = 4$

$t_i - y_i = y_{i+1} - t_i$, y_2 is the centroid of shaded figure

Nonuniform scalar quantizer

- The nonuniform quantizer has cells of different lengths.
- The cell lengths are small in the regions of highly probable values of random variable to be quantized.
- The cell lengths are large in the regions of lowly probable values of random variable to be quantized.

The Max-Lloyd procedure

1. Compute the number of cells $M = 2^{R_0}$

2. Compute thresholds $\{t_i\}$ for uniform quantizer. Set the current error value $D_c = E.$ Choose $\varepsilon > 0$ and the maximal number of iterations N_I.

3. Compute the approximating values $\{y_i\}, \quad i = 1,..., M$ according (2.2) and the new error value D according (2.4)

4. If $D_c - D > \varepsilon$ and the number of iterations is less than or equal to N_I compute the new thresholds $\{t_i\}$ $i = 1,..., M$ using (2.1), set $D_c = D$ and go to step 3, otherwise stop.

Max-Lloyd procedure

If pdf $f(x)$ is unknown then the same procedure is a applied to a sequence $x_1, x_2, ..., x_k$ of values of random variable X observed at the quantizer input during k sequential time moments.

In this case instead of formula (2.2) we use

$$y_i = \frac{\sum_{x \in [t_{i-1}, t_i)} x}{k_i},$$

where k_i is the number of $x \in (t_{i-1}, t_i]$.

M-L procedure results in reducing intervals which contain many values x_i and increasing intervals which contain small number of values x_i.

Entropy-constrained scalar quantizer

Variable-rate optimal nonuniform quantizer has the optimal number of cells, the optimal sets of approximating values and thresholds minimizing average distortions with an entropy constraint.

For this case the optimization problem reduces to the problem of finding the conditional minimum of MSE

$$D_{opt}(Q,R)=\min_{\{t_i\},\{y_i\},M}\left\{\sum_{i=1}^{M}\int_{t_{i-1}}^{t_i}(x-y_i)^2\,f(x)dx\right\},\ H(\{t_i\},M)=R_0.$$

To do that we construct the Lagrangian function

$$L=\left\{\sum_{i=1}^{M}\int_{t_{i-1}}^{t_i}(x-y_i)^2\,f(x)dx\right\}+\lambda\varphi(\{t_i\},M),\quad \varphi(\{t_i\},M)=H(\{t_i\},M)-R_0$$

and search for the minimum of L over $\{t_i\},\{y_i\},M$.

Entropy-constrained quantizer

The solution of optimization problem represents an iterative procedure.

We choose λ from a given range.

If $\lambda = const$ differentiating $L(\{t_i\},\{y_i\},\lambda)$ over t_i yields

$$(t_i - y_i)^2 - (t_i - y_{i+1})^2 - \lambda \log \frac{P(y_i)}{P(y_{i+1})} = 0, \quad i = 1,\ldots,M \qquad (2.5)$$

$$y_i = \frac{1}{P(y_i)} \int_{t_{i-1}}^{t_i} x f(x) dx,$$

$$P(y_i) = \int_{t_{i-1}}^{t_i} f(x) dx.$$

$t_0 = A \to t_1 \to P(y_1), y_1 \to t_2 \to P(y_2), y_2 (2.5) \to \ldots$ if $t_M \neq B$ new $t_1,\ldots$

Suboptimal scalar quantizers

1. Optimal uniform quantizer $t_i = \delta(i + 1/2),$

$$y_i = \frac{m_i}{P(y_i)}, \quad i = 1,\ldots,M$$

2. Quantizer with extended zero zone

$$t(j,\alpha) = \{\pm\alpha 2^{j-1}, \pm\alpha\left(2^{j-1}+1\right), \pm\alpha(2^{j-1}+2), \ldots\}$$

2.1 Optimal quantizer with EZZ

$$y_i = \frac{m_i}{P(y_i)}, \quad t(j,\alpha)$$

2.2 Suboptimal quantizer with EZZ

Vector quantization

Let $x \in X$ and X^n denote a set of vectors $\mathbf{x} = (x_1, ..., x_n)$ where x_i is value of x at time i.

- *Vector quantization* is a mapping Q of an input vector $\mathbf{x} = (x_1, ..., x_n)$ from X^n into the closest (with respect to the chosen distortion measure) approximating vector $\mathbf{y} = (y_1, ..., y_n)$ from the discrete *approximating set* $Y = \{\mathbf{y}_1, ..., \mathbf{y}_M\}$.

- *To construct a vector quantizer* means to split X^n into M areas or cells S_i such that $\bigcup_i S_i = X^n, S_i \bigcap S_j = \varnothing, \quad i \neq j,$ and $\mathbf{y}_i \in S_i$.

Vector quantization

n is dimension of the quantizer.

The quality of the quantizer is measured by the average quantization error due to replacing $\mathbf{x}$ by $\mathbf{y} = Q(\mathbf{x})$

$$D_n(Q) = E\{d(\mathbf{x}, Q(\mathbf{x}))\},$$

where $d(\mathbf{x}, Q(\mathbf{x}))$ is a nonnegative function called distortion measure or fidelity criterion. The most commonly used distortion measure:

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{n}\|\mathbf{x} - \mathbf{y}\|^2 = \frac{1}{n}(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T = \frac{1}{n}\sum_{j=1}^{n}(x_j - y_j)^2.$$

Vector quantization

$$D_n(Q) = \frac{1}{n} E\{\|\mathbf{x} - Q(\mathbf{x})\|^2\} = \frac{1}{n} \sum_{i=1}^{M} E\{\|\mathbf{x} - \mathbf{y}_i\|^2\}.$$

Assume that the n dimensional pdf $f(\mathbf{x})$ is known

$$D_n(Q) = \frac{1}{n} \sum_{i=1}^{M} \int_{S_i} f(\mathbf{x}) \|\mathbf{x} - \mathbf{y}_i\|^2 d\mathbf{x}$$

and the probability of the vector $\mathbf{y}_i$ is:

$$P(\mathbf{y}_i) = \int_{S_i} f(\mathbf{x}) d\mathbf{x}.$$

Vector quantization

The quantization rate is the number of bits required to represent the vector $\mathbf{x}$ per quantizer dimension n.

For the fixed-rate quantizer:

$$R = \frac{\log_2 M}{n} \, bits / sample.$$

For the variable-rate quantizer we can achieve

$$R = -\frac{1}{n} \sum_{j=1}^{M} P(\mathbf{y}_j) \log_2 P(\mathbf{y}_j).$$

Example

Let approximating set Y contain 4 vectors:

$$\mathbf{y}_1 = (0.2,\ 0.3)$$
$$\mathbf{y}_2 = (0.3,\ 0.1)$$
$$\mathbf{y}_3 = (0.1,\ 0.5)$$
$$\mathbf{y}_4 = (0.0,\ 0.4)$$

We quantize the vector $\mathbf{x} = (0.18,\ 0.25)$.

The closest approximating vector is $\mathbf{y}_1 = (0.2,\ 0.3)$

The average quantization error is equal to 1.45×10^{-3}.

The quantization rate is equal to

$$R = (\log_2 4)/2 = 1 \text{ bit/sample.}$$

Optimal vector quantization

If $f(\mathbf{x})$ is known then we can write MSE of the quantizer

$$D_n(Q) = \frac{1}{n} E\left\{\left\|\mathbf{x} - Q(\mathbf{x})\right\|^2\right\} = \frac{1}{n}\sum_{j=1}^{M}\int_{S_j}\left\|\mathbf{x} - \mathbf{y}_j\right\|^2 f(\mathbf{x})d\mathbf{x}$$

Differentiating with respect to $\mathbf{y}_i$ and setting derivatives to 0 we obtain

$$\mathbf{y}_i = \frac{\int_{S_i}\mathbf{x}f(\mathbf{x})d\mathbf{x}}{\int_{S_i} f(\mathbf{x})d\mathbf{x}}, \qquad i = 1,\ldots,M$$

are centroids of S_i. In general case it is impossible to find borders of S_i and $f(\mathbf{x})$ is usually unknown.

The Voronoi partition

Consider a discrete set $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \}$, where $\mathbf{p}_i \in R^n$.

Each point $\mathbf{p}_i \in R^n$ is a centroid of its *Voronoi region:*

$$V_i = \left\{ \mathbf{x} \in R^n : d(\mathbf{x}, \mathbf{p}_i) \leq d(\mathbf{x}, \mathbf{p}_j), \text{ for all } i \neq j. \right\}$$

The Voronoi regions never intersect (except at the boundaries). The union of the Voronoi regions coincides with R^n.

In the general case the Voronoi cells are not polytopal or even convex. If the distortion measure is squared error they are polytopal and borders of the cells can be specified from values of $\mathbf{p}_i$!

The Voronoi partition

For the given $\mathbf{x}$ we choose such $\mathbf{y}_i \in Y$, that the following inequality holds $\|\mathbf{x}-\mathbf{y}_i\|^2 \le \|\mathbf{x}-\mathbf{y}_j\|^2, \quad i \ne j.$

The border between S_i and S_j is a hyperplane perpendicular to the segment connecting $\mathbf{y}_i$ and $\mathbf{y}_j$:

$$H_{ij} = \left\{ \mathbf{x} : (\mathbf{x}, (\mathbf{y}_j - \mathbf{y}_i) + \frac{1}{2}\left(\|\mathbf{y}_i\|^2 - \|\mathbf{y}_j\|^2 \right) = 0 \right\}.$$

Every hyperplane that determines a border of a cell is characterized by two approximating vectors to which this hyperplane is equidistant. The set $\{\mathbf{y}_i\}$ uniquely determines the shapes of the quantization cells.

The Linde-Buzo-Gray algorithm

Let $\mathbf{x}_1 = (x_{11},..., x_{1n}), \mathbf{x}_2 = (x_{21},..., x_{2n}),..., \mathbf{x}_k = (x_{k1},..., x_{kn})$ be a sequence of vectors observed at the input of the quantizer. Choose $M = 2^{R_0 n} << k, \quad \varepsilon > 0, \quad N_I.$

1. Initialize the codebook by choosing as $\mathbf{y}_i,\ i = 1,..., M$ arbitrary vectors $\mathbf{x} \in \{\mathbf{x}_j, j = 1,.., k\}$, $D_p = \infty.$
2. Set $\mathbf{s}_i = (0,...,0),\ N_i = 0$, $i = 1,..., M,\quad \mathrm{D}_c = 0.$
   For each $\mathbf{x}_j,\ j = 1,.., k$ find the closest codeword $\mathbf{y}_i.$
   Modify auxiliary variables $\mathbf{s}_i = \mathbf{s}_i + \mathbf{x}_j,\ N_i = N_i + 1,$
   $D_c = D_c + \frac{1}{n}\left\|\mathbf{x}_j - \mathbf{y}_i\right\|^2$
3. Update the codebook $\mathbf{y}_i = \mathbf{s}_i / N_i$, $i = 1,..., M$
4. Update the average error $D_c = D_c / k$
5. Stop if $D_p - D_c < \varepsilon$ or number of iterations $> N_I$
   otherwise $D_p = D_c$ and go to step 2.

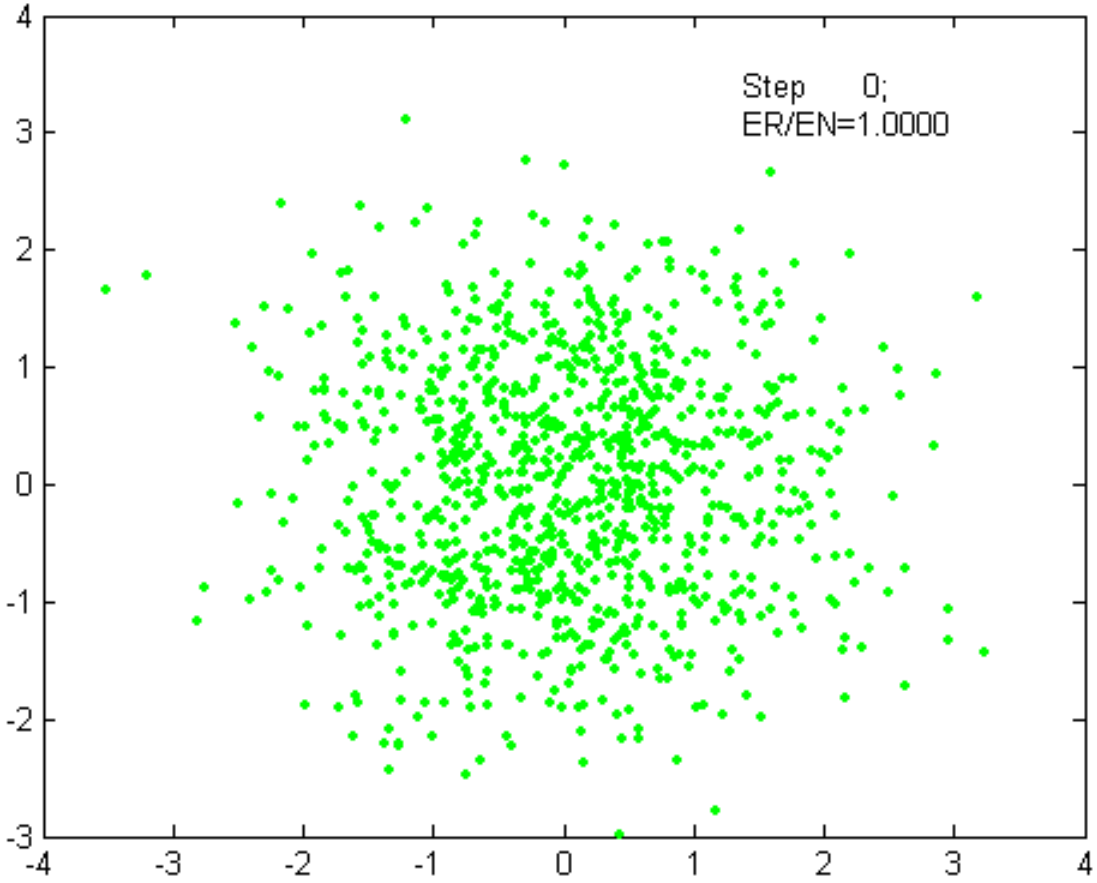
The Linde-Buzo-Gray algorithm

The main shortcoming of vector quantization is its high computational complexity. To find the closest approximating vector it is necessary to fulfil an exhaustive search among $M = 2^{nR_0}$ vectors.

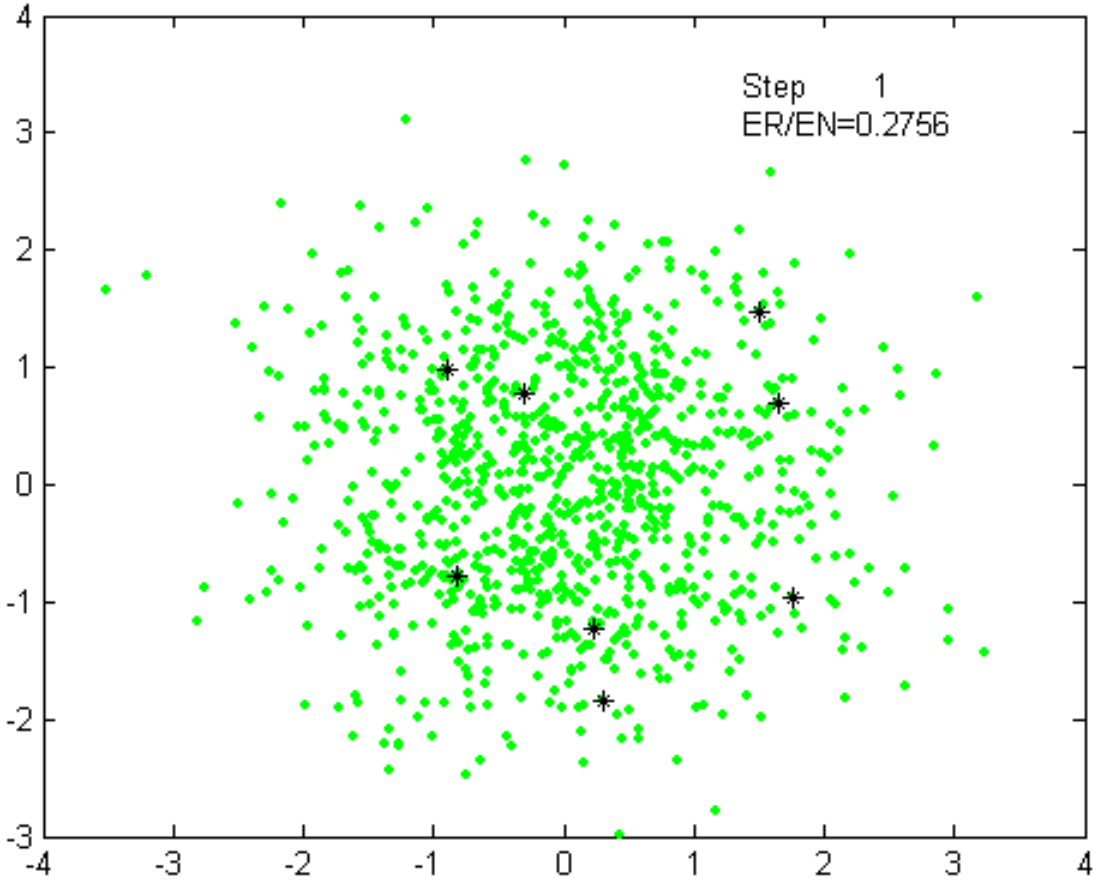
For example, if $R_0 = 0.5$ bits/sample and $n = 32$ the size of the book is equal to $M = 2^{16}$.

It is possible to reduce the computational complexity by using structured codebooks. We can use trellis codes as codebooks. In this case searching for the closest codeword can be organized by the Viterbi algorithm.

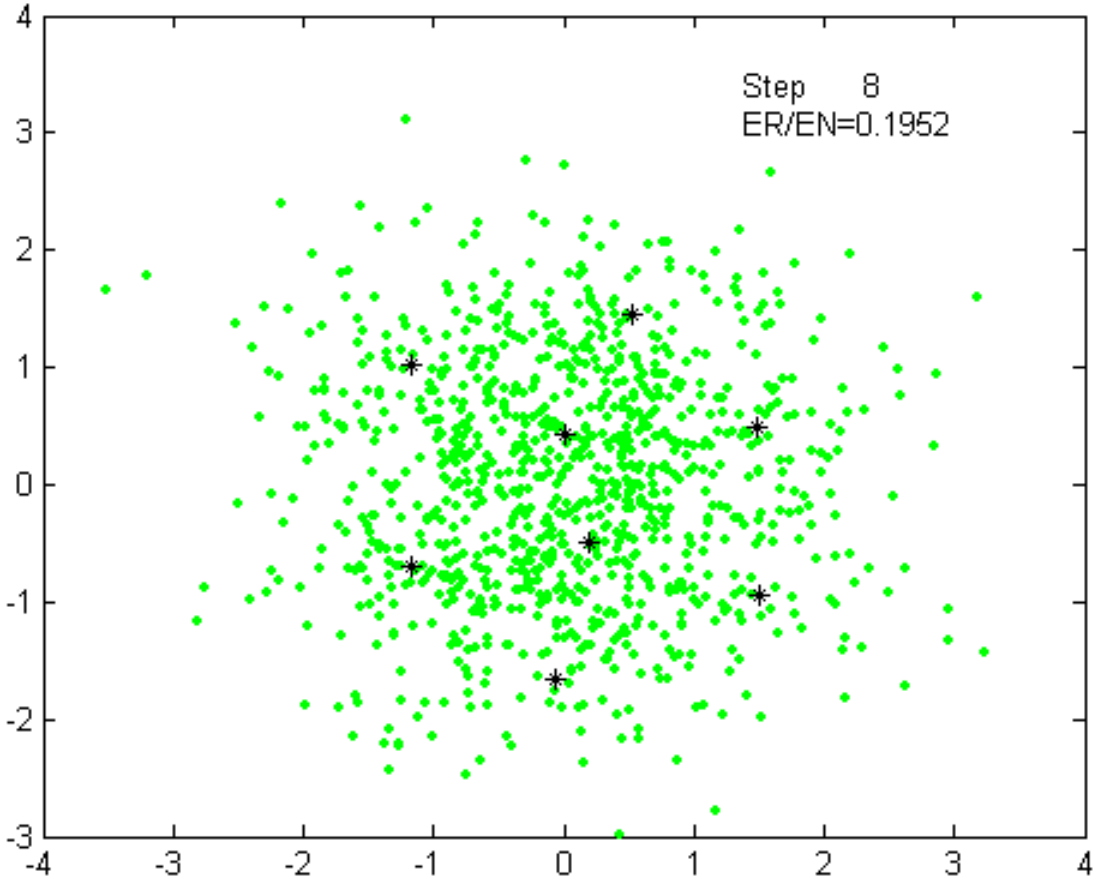
LBG procedure



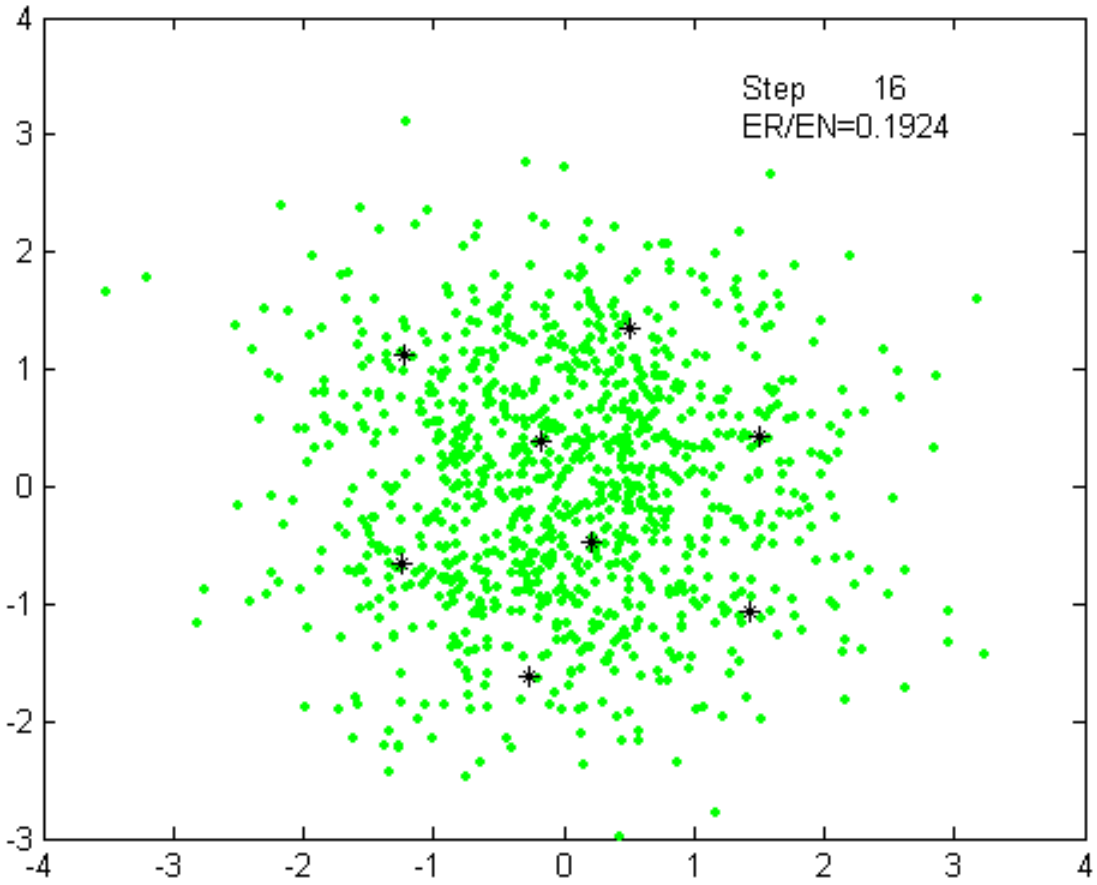
LBG procedure



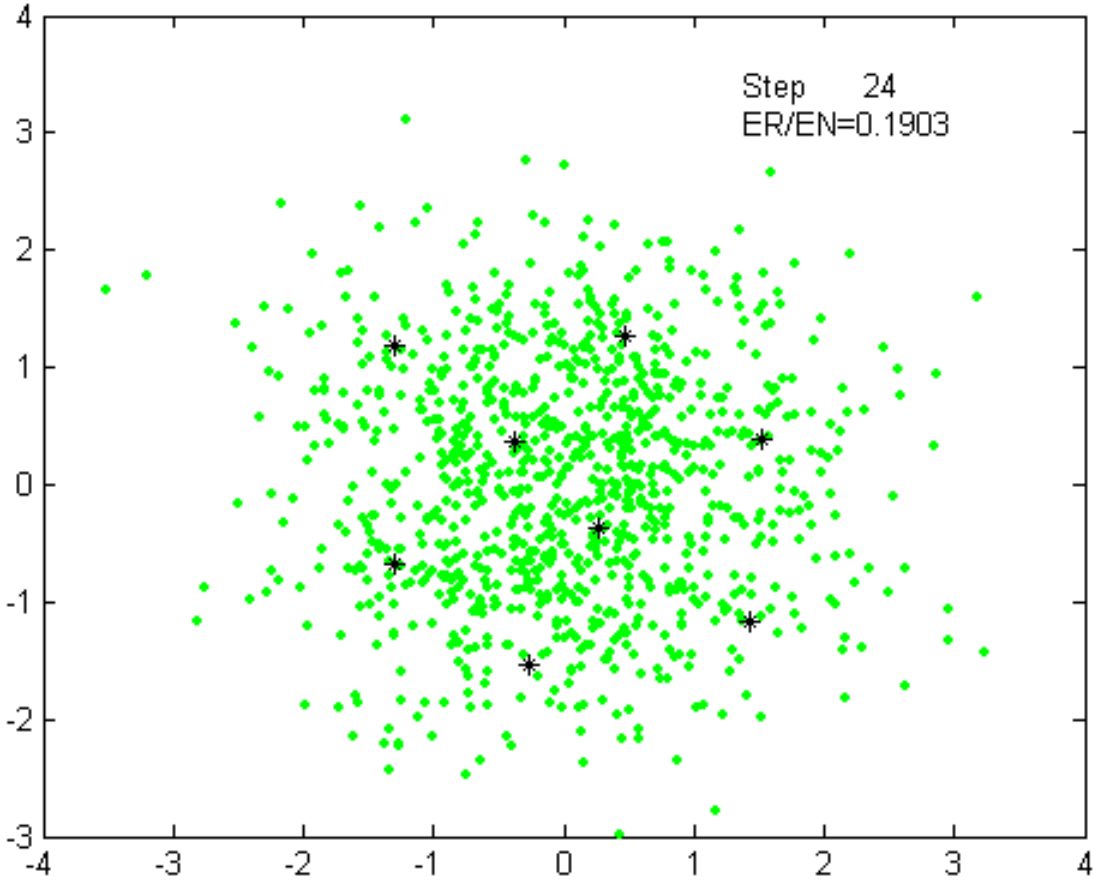
LBG procedure



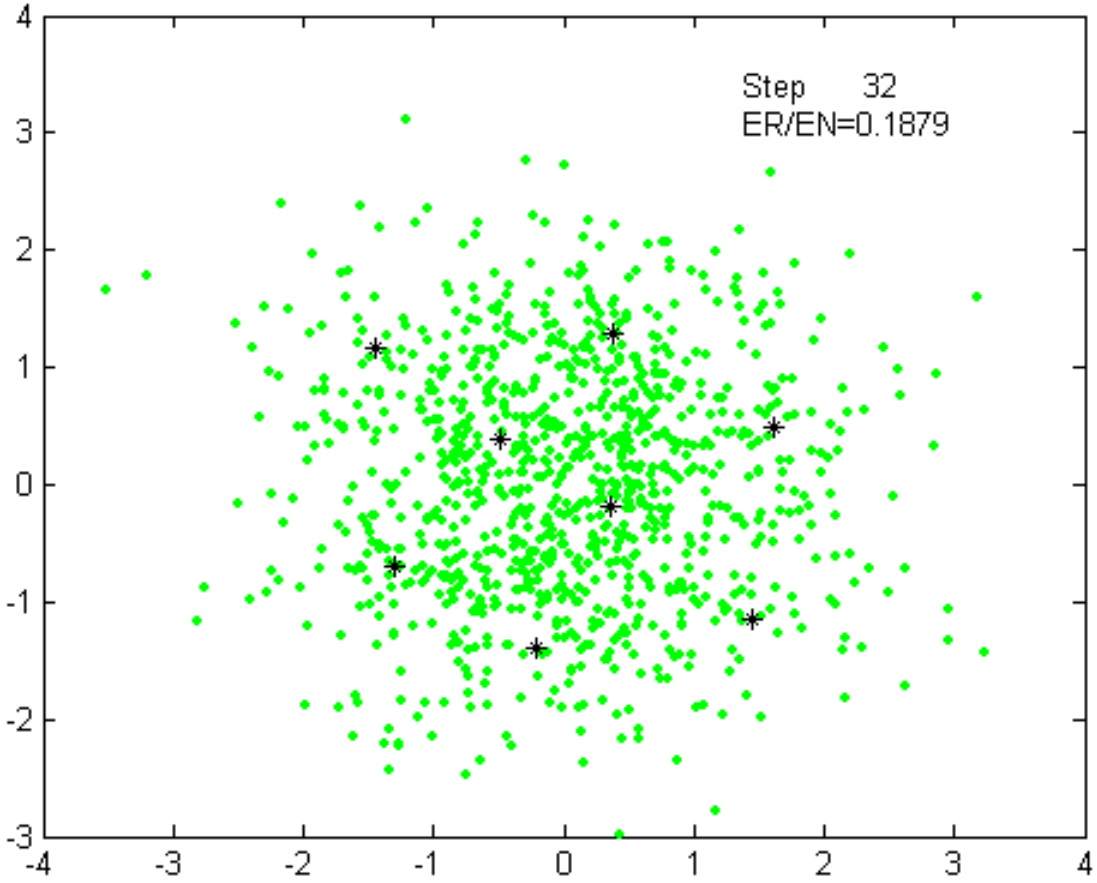
LBG procedure



LBG procedure



LBG procedure



LBG procedure