Lossy Compression

Quantization, Transform Coding

October 2018

Content

- Quantization
- Transform Coding

Quantization

Quantization examples:

- Rounding and truncating:
  $x \in \mathbb{R} \longrightarrow y = Q(x) = round(x) \in \mathbb{Z}$
- A/D (D/A) Converter: $x \in \mathbb{R} \longrightarrow y =$ n-bit integer

Formally,

$$x \in \mathbb{S} \longrightarrow y = Q(x) \in \mathbb{D} \tag{1}$$

Where:

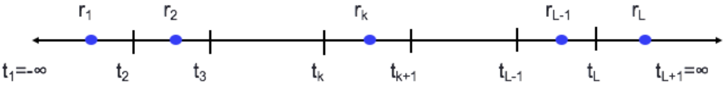
- $\mathbb{S}$, source space, is a continous space (or discrete but very large) space
- $\mathbb{D}$, destination space, is a discrete space, $\mathbb{D} \subset \mathbb{Z}$

Quantizer

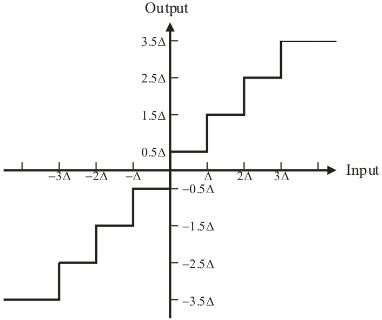
Quantizer $=$ *encoder* mapping $+$ *decoder* mapping

Quantizer's encoder mapping $y = Q(x)$:

- devides source space into multiple intervals,
- if x falls in interval k, $y = Q(x) = r_k$
- Code $\{r_k\}$: fix length codes or variable length codes
- $\longrightarrow$ is **irreversible**



Quantizer (cont.)



- $\{r_k\}$: *reconstruction levels*
- $\{t_k\}$: *transition levels* or *decision boundaries*
- $\Delta_k = t_{k+1} - t_k$: *quantization step*
- $x - Q(x)$: *quantization error* $\rightarrow$ *MSE*
- *Rate of quantizer*: rate of coding scheme for r_k
- **midrise** and **midread** quantizer

Quantizer design approaches

- fixed-length intervals, fixed-length code
- **variable-length intervals, fixed-length code**
- fixed-length intervals, variable-length code
- variable-length intervals, variable-length code

Rate and distortion of quantizers

Rate:

$$
\begin{aligned}
R &= \sum_{i=1}^{M} l_i P(r_i) \\
&= \sum_{i=1}^{M} l_i \int_{t_{i-1}}^{t_i} f(x) dx
\end{aligned}
$$

Distortion:

$$
\begin{aligned}
D &= E[(x - Q(x))^2] \\
&= \sum_{i=1}^{M} \int_{t_{i-1}}^{t_i} (x - Q(x))^2 f(x) dx
\end{aligned}
$$

Quantizer design problem

Given an input signal X with $pdf = f(x)$. How to design a quantizer with M levels such that:

- Distortion is minimizied while satisfied rate constraint $R \leq R^*$
- Rate is minimized while satisfied distortion contrain $\sigma^2 \leq D^*$

Simplified (and typical) form: *fixed-length code, fixed number of steps*

- Given an input signal X with $pdf = f(x)$. How to design a quantizer with M levels such that distortion is minimizied

Lloyd-Max quantizer design algorithm

- Centroid condition: *If* t_i *are fixed, best* r_i *should be "center of gravity" of each intervals*

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

- Nearest neighborhood condition: *If* r_i *are fixed, best* t_i *should be center of* $[r_{i-1}, r_i]$

$$t_i = \frac{1}{2}(r_{i-1} + r_i)$$

Uniform quantizer

Given signal $X \in [min, max]$, with pdf $f(x) = \frac{1}{max - min}$ and a $M - level$ uniform quantizer, calculate the followings:

- Quantization steps Δ
- Quantization's transition levels $\{t_k | k = 0..M\}$
- Optimum quantization's reconstruction levels $\{r_k | k = 0..M-1\}$
- Quantization error

Non-uniform quantizer

Given signal X, with pdf $f(x)$, find an optimal $M - level$ non-uniform quantizer:

- Quantization's transition levels $\{t_k | k = 0..M\}$??
- Optimum quantization's reconstruction levels $\{r_k | k = 0..M-1\}$??

Answer:

- Lloyd-Max interative algorithm
- Inverse function method (Probability Integral transform)

Probability Integral Transform

If X has distribution (cdf) $F_X(x)$, $Y = F_X(X)$ is a uniformly distributed ranvar
Proof:

$$
\begin{aligned}
F_Y(Y) &= P(Y \leq y) = P(F_X(X) \leq y) \\
&= P(X \leq F_X^{-1}(y)) = F_X(F_X^{-1}(y)) \\
&= y
\end{aligned}
$$

Application:

- Transform X to Y using Probabbility Integral Transform $F_X(.)$
- Design uniform quantizer $Q_Y(.)$ for Y
- Transform $Q_Y(.)$ using F_X^{-1} to get $Q_X(.)$

Supplementary: How to generate random variables with a specific distribution?

Q: How to generate ranvar Y with cdf $F_Y(y)$ from uniformly distributed ranvar X?
A: If $F_Y(.)$ is invertable, take $Y = F_Y^{-1}(X)$

Q: How to generate ranvar $Y \in \{y_k\}$ whose distribution are p_k?
A1: Segment unit interval into sub-intervals. Each interval is assigned a symbol y_k. Generate x from $uniform(0, 1)$. If x falls in a sub-interval, output the corresponding symbol y_k
A2: Arithmetic decoder

Transform Coding

Transform Coding: Transform origin signals before samples dropping or quantization

Typical transforms:

- Discrete Cosin Transform - DCT
- Single Value Decomposition - SVD
- Wavelet
- Prediction transforms
- Model fitting

Why?:

- Transformation may yield more efficient representation of the original samples
- Transformed signal may require fewer bit to code.

Discrete Cosine Transform

Can be considered as "real"version of DFT:

DFT

$$\mathbf{X} = \mathbf{F}\mathbf{x}$$

$$\begin{aligned}\mathbf{F} &= \frac{1}{\sqrt{N}} \begin{pmatrix} \cdots \\ e^{j\frac{\pi}{N}2kn} \\ \cdots \end{pmatrix} \\ &= \frac{1}{\sqrt{N}} \begin{pmatrix} \cdots \\ cos(\frac{\pi}{N}2kn) \\ \cdots \end{pmatrix} \\ &+ j\frac{1}{\sqrt{N}} \begin{pmatrix} \cdots \\ sin(\frac{\pi}{N}2kn) \\ \cdots \end{pmatrix}\end{aligned}$$

DCT

$$\mathbf{X} = \mathbf{C}\mathbf{x}$$

$$\mathbf{C} = \alpha(k) \begin{pmatrix} \cdots \\ cos(\frac{\pi}{N}\frac{2n+1}{2}k) \\ \cdots \end{pmatrix}$$

$$\begin{aligned}\alpha(k) &= \sqrt{\frac{1}{N}}, k = 0 \\ \alpha(k) &= \sqrt{\frac{2}{N}}, k > 0\end{aligned}$$

DCT

Matrix version of DCT

$$C_1 = \sqrt{\frac{2}{N}} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ cos(\frac{\pi}{2N}) & cos(\frac{3\pi}{2N}) & \dots & cos(\frac{(2N-1)\pi}{2N}) \\ cos(2\frac{\pi}{2N}) & cos(2\frac{3\pi}{2N}) & \dots & cos(2\frac{(2N-1)\pi}{2N}) \\ \dots & \dots & \dots & \dots \\ cos((N-1)\frac{\pi}{2N}) & cos((N-1)\frac{3\pi}{2N}) & \dots & cos((N-1)\frac{(2N-1)\pi}{2N}) \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 1 & 1 & \dots & 1 \\ cos(\frac{\pi}{2N}) & cos(\frac{3\pi}{2N}) & \dots & cos(\frac{(2N-1)\pi}{2N}) \\ cos(2\frac{\pi}{2N}) & cos(2\frac{3\pi}{2N}) & \dots & cos(2\frac{(2N-1)\pi}{2N}) \\ \dots & \dots & \dots & \dots \\ cos((N-1)\frac{\pi}{2N}) & cos((N-1)\frac{3\pi}{2N}) & \dots & cos((N-1)\frac{(2N-1)\pi}{2N}) \end{pmatrix}$$

IDCT - Inverse DCT

$$
\begin{aligned}
C_1^{-1} &= \sqrt{\frac{2}{N}} \begin{pmatrix} \frac{1}{\sqrt{2}} & cos(\frac{\pi}{2N}) & \ldots & cos((N-1)\frac{\pi}{2N}) \\ \frac{1}{\sqrt{2}} & cos(\frac{3\pi}{2N}) & \ldots & cos((N-1)\frac{3\pi}{2N}) \\ \ldots & \ldots & \ldots & \ldots \\ \frac{1}{\sqrt{2}} & cos(\frac{(2N-1)\pi}{2N}) & \ldots & cos((N-1)\frac{(2N-1)\pi}{2N}) \end{pmatrix} \\
&= C_1^T
\end{aligned}
$$

$$
C_2^{-1} = \frac{2}{N} \begin{pmatrix} \frac{1}{2} & cos(\frac{\pi}{2N}) & \ldots & cos((N-1)\frac{\pi}{2N}) \\ \frac{1}{2} & cos(\frac{3\pi}{2N}) & \ldots & cos((N-1)\frac{3\pi}{2N}) \\ \ldots & \ldots & \ldots & \ldots \\ \frac{1}{2} & cos(\frac{(2N-1)\pi}{2N}) & \ldots & cos((N-1)\frac{(2N-1)\pi}{2N}) \end{pmatrix}
$$

DCT - illustration

$$\begin{array}{cc} & \xrightarrow{\qquad\qquad j \qquad\qquad} \\ i \Bigg\downarrow & \begin{pmatrix} \cdots \\ cos\left((2j+1)i\frac{\pi}{N}\right) \\ \cdots \end{pmatrix} \end{array}$$

4-point DCT

$$C(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ cos(\frac{\pi}{8}) & cos(3\frac{\pi}{8}) & cos(5\frac{\pi}{8}) & cos(7\frac{\pi}{8}) \\ cos(2\frac{\pi}{8}) & cos(6\frac{\pi}{8}) & cos(10\frac{\pi}{8}) & cos(14\frac{\pi}{8}) \\ cos(3\frac{\pi}{8}) & cos(9\frac{\pi}{8}) & cos(15\frac{\pi}{8}) & cos(21\frac{\pi}{8}) \end{pmatrix}$$