Computational Approach to Lossy Compression and its Applications to Image/Video Coding

En-hui Yang

University of Waterloo

Formula for Spending Big Money in a Big Way

Neat formulas

$\Downarrow$

Dirty formulas

$\Downarrow$

Protecting dirty formulas

$\Downarrow$

Transforming dirty formulas

$\Downarrow$

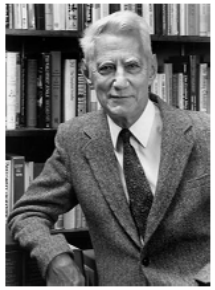
Commercializing dirty formulas

Outline

Outline

Rate Distortion Theory in Shannon Sense

One source, one encoder, and one decoder (3O compression)



- Pioneered by Shannon in 1948 and particularly, in 1959.
- Probabilistic model: $X_1X_2\cdots X_n\cdots$ is modeled as an IID source or in general a stationary, ergodic source.
- No constraints whatsoever on the complexity/capability of the encoder and decoder.
- The best rate distortion tradeoff is neatly characterized by

$$R(D) = \inf_{\hat{X}: Ed(X,\hat{X}) \le D} I(X;\hat{X}) \text{ in the IID case}$$

and, in the general stationary ergodic case, by

$$R(D) = \lim_{n\to\infty} \frac{1}{n} \inf_{\hat{X}^n: Ed(X^n,\hat{X}^n) \le D} I(X^n;\hat{X}^n).$$

Impact/Limitation

- High expectations: believed to provide in principle a theoretical basis for many practically important lossy compression problems.
- Big disappointments: Impacts on practice? Where?
- Why?
  - The fundamental modeling is problematic: real-world data are often nonstationary and may not fit into any analytical model; even if they do, such a model is very difficult to construct.
  - Asymptotic analysis is misleading: with asymptotic analysis, the impact of lossless coding and the selection of reproduction space on the overall lossy compression performance is ignored.

Outline

General Lossy Codes

Notation

- $\mathcal{X}$: our source alphabet; in some toy examples, $\mathcal{X}$ could be finite; in practice, $\mathcal{X}$ is often the real line $(-\infty, +\infty)$.
- $\hat{\mathcal{X}}$: a reproduction alphabet; in some toy examples, $\hat{\mathcal{X}}$ could be finite and different from $\mathcal{X}$; in practice, $\hat{\mathcal{X}}$ is often the same as $\mathcal{X} = (-\infty, +\infty)$.
- $d : \mathcal{X}^n \times \hat{\mathcal{X}}^n \to [0, \infty)$: a distortion measure indicating the quality loss per symbol when $\hat{x} \in \hat{\mathcal{X}}^n$ is reproduced at the decoder side to represent $x \in \mathcal{X}^n$. d is additive if

$$d(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i) \tag{1}$$

for any $x = x_1 x_2 \cdots x_n$ and $\hat{x} = \hat{x}_1 \hat{x}_2 \cdots \hat{x}_n$, $n \geq 1$.

General Lossy Codes

Notation

- $\mathcal{X}$: our source alphabet; in some toy examples, $\mathcal{X}$ could be finite; in practice, $\mathcal{X}$ is often the real line $(-\infty, +\infty)$.
- $\hat{\mathcal{X}}$: a reproduction alphabet; in some toy examples, $\hat{\mathcal{X}}$ could be finite and different from $\mathcal{X}$; in practice, $\hat{\mathcal{X}}$ is often the same as $\mathcal{X} = (-\infty, +\infty)$.
- $d : \mathcal{X}^n \times \hat{\mathcal{X}}^n \to [0, \infty)$: a distortion measure indicating the quality loss per symbol when $\hat{x} \in \hat{\mathcal{X}}^n$ is reproduced at the decoder side to represent $x \in \mathcal{X}^n$. d is additive if

$$d(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i) \tag{1}$$

for any $x = x_1 x_2 \cdots x_n$ and $\hat{x} = \hat{x}_1 \hat{x}_2 \cdots \hat{x}_n$, $n \geq 1$.

General Lossy Codes

Notation

- $\mathcal{X}$: our source alphabet; in some toy examples, $\mathcal{X}$ could be finite; in practice, $\mathcal{X}$ is often the real line $(-\infty, +\infty)$.
- $\hat{\mathcal{X}}$: a reproduction alphabet; in some toy examples, $\hat{\mathcal{X}}$ could be finite and different from $\mathcal{X}$; in practice, $\hat{\mathcal{X}}$ is often the same as $\mathcal{X} = (-\infty, +\infty)$.
- $d : \mathcal{X}^n \times \hat{\mathcal{X}}^n \to [0, \infty)$: a distortion measure indicating the quality loss per symbol when $\hat{x} \in \hat{\mathcal{X}}^n$ is reproduced at the decoder side to represent $x \in \mathcal{X}^n$. d is additive if

$$d(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i) \tag{1}$$

for any $x = x_1 x_2 \cdots x_n$ and $\hat{x} = \hat{x}_1 \hat{x}_2 \cdots \hat{x}_n$, $n \geq 1$.

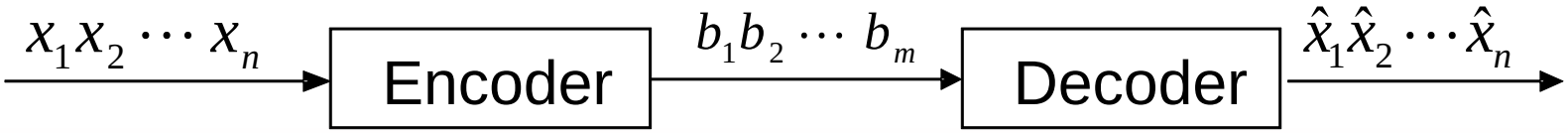


Figure: 1 Diagram of lossy compression

Definition

The notion of a lossy code is illustrated in Figure 1. Mathematically, a lossy code is a pair $\mathcal{C} = (f, g)$, where f is a mapping from $\mathcal{X}^+$ to $\mathcal{B}^+$, acting as an encoder, g is a partial mapping from $\mathcal{B}^+$ to $\hat{\mathcal{X}}^+$, acting as a decoder, and for any sequence $x \in \mathcal{X}^+$,

$$\hat{x} = g(f(x))$$

is well defined and has the same length as x. The performance of $\mathcal{C} = (f, g)$ on each $x \in \mathcal{X}^+$ is measured by its rate (in bits per symbol) $|f(x)|/|x|$ and distortion (per symbol) $d(x, \hat{x})$.

Assume that there is no transmission loss between the encoder and decoder. Then the decoder g is more or less a robot. As such, the encoder f in Figure 1 actually has two important steps: 1) compute $\hat{x}$ from x, and 2) encode either directly or indirectly $\hat{x}$ in a lossless manner. Since in practical lossy compression, $\hat{x}$ is discrete while x is continuous, the first step is broadly referred to as *quantization*. Accordingly, as shown in Figure 2, a lossy code can be also defined alternatively as follows.

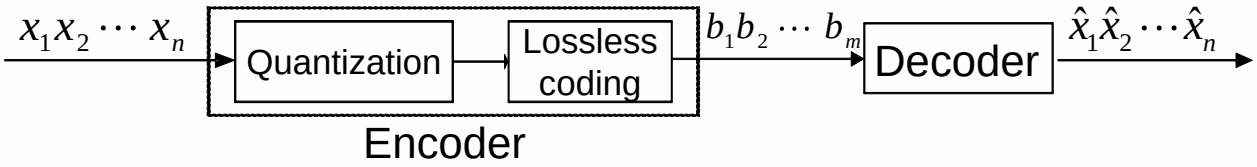


Figure: 2 Alternative diagram of lossy compression

Alternative Definition

a lossy code is a triple $\mathcal{C} = (Q, \phi, g)$, where

- $Q = (Q_1, Q_2)$ represents a quantization process which, for each $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$, $n \geq 1$, first computes a reproduction space $Q_1(x) \subset \hat{\mathcal{X}}^n$ and then selects a reproduction sequence $Q_2(x) \in Q_1(x)$;
- ϕ encodes both $Q_1(x)$ and $Q_2(x)$ into a binary codeword;
- Q and ϕ together act as an encoder; and
- g is a partial mapping from $\mathcal{B}^+$ to $\hat{\mathcal{X}}^+$, acting as a decoder, such that

$$g(\phi(Q_1(x), Q_2(x))) = Q_2(x)$$

for each sequence $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$ and for any $n \geq 1$.

The performance of $\mathcal{C} = (Q, \phi, g)$ on x is once again measured by its rate (in bits per symbol) $|\phi(Q_1(x), Q_2(x))|/n$ and distortion (per symbol) $d(x, Q_2(x))$.

In comparison with standard lossy codes considered in rate distortion theory in Shannon sense such as scalar quantizers, vector quantizers, and trellis quantizers, the following two distinctions stand out:

- The reproduction space $Q_1(x) \subset \hat{\mathcal{X}}^n$ is generally sequence dependent and hence has to be sent to the decoder; its size could also be countably infinite. This is pretty common in practice, but more or less neglected in Shannon's probabilistic approach to rate distortion theory.
- The quantization step Q actually involves the concepts of both space and time: $Q_1(x)$ represents space and $Q_2(x)$ represents time. This space-time perspective to lossy compression is yet to be fully explored in both theory and practice.

Special Case 1: Scalar Quantizers

A lossy code $\mathcal{C} = (Q, \phi, g)$ is said to be a *scalar quantizer* if

- there is a finite set $\mathcal{Y} = \{y_1, \cdots, y_L\} \subset \hat{\mathcal{X}}$ such that for any $n \geq 1$ and any $x = x_1 \cdots x_n \in \mathcal{X}^n$, $Q_1(x) = \mathcal{Y}^n$, and
- for any $n \geq 1$ and any $x = x_1 \cdots x_n \in \mathcal{X}^n$,

$$Q_2(x) = Q_2(x_1)Q_2(x_2)\cdots Q_2(x_n). \tag{2}$$

In this case, the entire quantization process can be specified by the mapping $Q_2 : \mathcal{X} \to \mathcal{Y} = \{y_1, \cdots, y_L\}$, and one can simply identify Q_2 with Q. $\mathcal{C}$ is said to be of *fixed rate* if each $Q_2(x_i)$ is represented by $\lceil \log L \rceil$ bits, and of *variable rate* otherwise.

Example: Symmetric Uniform Quantizers of the Midtread Type

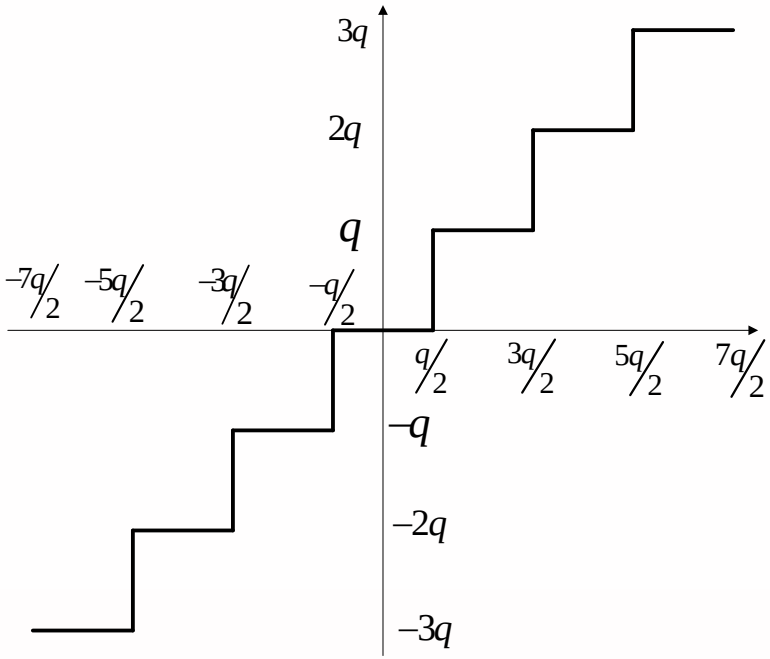


Figure: 3 A symmetric uniform quantizer of the midtread type

- $Q(x) = q \times \text{round}\left(\frac{x}{q}\right)$
- Used in JPEG and all video coding standards proposed so far.

Hard Decision Quantization (HDQ)

The quantization process given in (2) is also called *hard decision quantization* since the reproduction symbol corresponding to each source symbol x_i is uniquely determined by the source symbol x_i only.

Soft Decision Quantization (SDQ)

When inter-symbol correlations are utilized in the lossless coding of $Q_2(x_1 x_2 \cdots x_n)$, $n > 1$, HDQ is in general not efficient in terms of the rate distortion trade-off between the rate $|\phi(Q_1(x), Q_2(x))|/n$ and the distortion $d(x, Q_2(x))$. In this case, a better way of performing quantization is through a technique called *soft decision quantization*. Given the reproduction space $Q_1(x)$ for each $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$, $n \geq 1$, in SDQ, one maps x into $Q_2(x) \in Q_1(x)$ in such a way that the ith symbol in $Q_2(x)$, $i = 1, 2, \cdots, n$, can not be determined in general from x_i only.

Outline

Rate Distortion Optimization

Let $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$ be a sequence to be compressed.

- $x = x_1 x_2 \cdots x_n$ is regarded as a deterministic sequence; there is no need for modeling.
- In view of the alternative definition of lossy code $\mathcal{C} = (Q, \phi, g)$, the best rate distortion performance of $x \in \mathcal{X}^n$ can be expressed by the following optimization

$$\inf_{\phi, Q_1(x)} \min_{Q_2(x)} \left[\frac{|e(\phi)| + |\phi(Q_1(x), Q_2(x))|}{n} + \lambda d(x, Q_2(x)) \right] \tag{3}$$

where $|e(\phi)|$ represents the number of bits needed to inform the decoder of ϕ, and $\lambda > 0$ represents the relative weights assigned to the rate and distortion.

- The infimum in (3) is taken over all possible ϕ and $Q = (Q_1, Q_2)$ without any constraints.
- Lossy coding given by solutions or approximate solutions to (3) is called *fixed slope lossy coding* since $-\lambda$ is corresponding to the slope of the rate distortion curve.
- If one requires that $d(x, Q_2(x)) \leq D$, then one can turn (3) into constrained optimization:

$$\inf_{\phi, Q_1(x)} \min_{Q_2(x)} \left[\frac{|e(\phi)| + |\phi(Q_1(x), Q_2(x))|}{n} \right] \tag{4}$$

subject to

$$d(x, Q_2(x)) \leq D.$$

Accordingly, lossy coding given by solutions or approximate solutions to (4) is called *lossy coding at a fixed distortion level*.

- Given ϕ and Q_1, lossy coding given by solutions or approximate solutions to the inner minimization of (3) is broadly referred to as *soft decision quantization (SDQ)*. SDQ is playing and will continue to play an important role in image and video coding standards (current and future).
- Equation (3) not only tightly couples lossy coding and lossless coding to the extent that SDQ can be regarded a technique of designing lossy coding algorithms from lossless algorithms, but also brings the quantization space into the joint optimization.
- In (3), one also has the flexibility to limit/control the decoding complexity/capability by putting some constraints on ϕ. This happens to be consistent with the design philosophy of all image/video coding standards developed so far.

Approximate Solution: General Alternating Algorithm

Given the reproduction space $Q_1(x) \subset \hat{\mathcal{X}}^n$, no matter how $Q_2(x)$ is selected from $Q_1(x)$, $Q_2(x)$ can be equally represented by its corresponding index sequence $U(x)$, and the lossless compression of $Q_2(x)$ given $Q_1(x)$ is equivalent to that of $U(x)$. As such, one can write

$$Q_2(x) = Q_3(Q_1(x), U(x)) \text{ and } \phi(Q_1(x), Q_2(x)) = \phi(Q_1(x), U(x)) \tag{5}$$

where $Q_3(\cdot,\cdot)$ is known and normally dictated by the structure of the reproduction space $Q_1(x)$.

Example 3.6: Illustration of $U(x)$ and Q_3

Let $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\}$ and $\mathcal{L} = \{1, 2, \cdot, L\}$. If $Q_1(x) = \mathcal{Y}^n$, then $U(x) = u_1 u_2 \cdots u_n$ would be an index sequence from $\mathcal{L}^n$ and the corresponding $Q_2(x)$ is equal to $y_{u_1} y_{u_2} \cdots y_{u_n}$.

In view of (5), we can rewrite (3) as follows:

$$\inf_{\phi, Q_1(x)} \min_{U(x)} \left[\frac{|e(\phi)| + |\phi(Q_1(x), U(x))|}{n} + \lambda d(x, Q_3(Q_1(x), U(x))) \right]. \tag{6}$$

The double minimization in (6) automatically renders an alternating algorithm for fixed slope lossy coding, at least in theory:

1: Initialize ϕ^0 and $Q_1^0(x)$.

2: Fix ϕ^{i-1} and $Q_1^{i-1}(x)$. Perform optimal SDQ to compute

$$U^i(x) = \arg\min_{U(x)} \left\{ \frac{|e(\phi^{i-1})| + |\phi^{i-1}(Q_1^{i-1}(x), U(x))|}{n} + \lambda d(x, Q_3(Q_1^{i-1}(x), U(x))) \right\}. \quad (7)$$

3: Fix $U^i(x)$. Compute

$$(\phi^i, Q_1^i(x)) = \arg\min_{\phi, Q_1(x)} \left\{ \frac{|e(\phi)| + |\phi(Q_1(x), U^i(x))|}{n} + \lambda d(x, Q_3(Q_1(x), U^i(x))) \right\}. \quad (8)$$

4: Repeat the above two steps for $i = 1, 2, \cdots$ until the convergence criteria are met, and then output $e(\phi^i)$ followed by $\phi^i(Q_1^i(x), U^i(x))$ as the compressed bitstream for x.

Remark

- The practicality of the above general alternating algorithm for fixed slope lossy compression depends on how difficult it is to find solutions or approximate solutions to (7) and (8), which in turn depends on the structure of the reproduction space $Q_1(x)$ and how complicate the lossless coding algorithm ϕ is. As we shall see later, for trellis reproduction spaces and lossless coding using run-length coding, Huffman coding, V2V coding, arithmetic coding, or any combination thereof, algorithms of relatively low complexity can be developed for finding solutions or approximate solutions to (7) and (8) in general.

- Even when optimal SDQ algorithms can be developed to solve (7), different optimal SDQ algorithms are required for different lossless coding algorithms ϕ, and the development of these optimal SDQ algorithms is sometimes quite challenging even when ϕ is run-length coding, Huffman coding, V2V coding, arithmetic coding, or any combination thereof.
- In some applications such as optimizing video/image coding standards without changing their respective syntax, the lossless coding algorithm ϕ is fixed. In this case, the alternating algorithm computes only $U^i(x)$ and $Q_1^i(x)$ in each iteration.

Example: product reproduction spaces and zero-order adaptive arithmetic coding

- $\mathcal{X} = \hat{\mathcal{X}} = (-\infty, +\infty)$.
- d: the squared error distortion measure.
- $\mathcal{L} = \{1, 2, \cdots, L\}$.
- $Q_1(x) = \mathcal{Y}^n$ for some $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\}$. In this case, one can ignore the bit rate arising from the encoding of $Q_1(x)$ in the optimization (6).
- ϕ encodes $U(x)$ using the zero-order adaptive arithmetic coding algorithm over $\mathcal{L}$.

For any $U(x) = u_1 u_2 \cdots u_n \in \mathcal{L}^n$,

$$\begin{aligned} |\phi(U(x))| &= \left\lceil -\log \frac{\prod_{l\in\mathcal{L}}(n_l)!}{n!\binom{n+L-1}{L-1}} \right\rceil + 1 \\ &= n\sum_{l=1}^{L}\left[-\frac{n_l}{n}\log\frac{n_l}{n}\right] + O(\log n) \end{aligned} \tag{9}$$

where n_l is the number times the index l appears in $U(x)$, the summation in (9) is the zero-order empirical entropy of $U(x)$, and the term $O(\log n)$ is generally independent of $U(x)$ and represents the cost of transmitting implicitly empirical frequencies n_l to the decoder by the adaptive arithmetic coding. On the other hand,

$$d(x, Q_3(Q_1(x), U(x))) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_{u_i})^2. \tag{10}$$

In view of (9) and (10), the object cost function in (6) can be simplified in this case as follows:

$$\sum_{l=1}^{L}\left[-\frac{n_l}{n}\log\frac{n_l}{n}\right]+\frac{\lambda}{n}\sum_{i=1}^{n}(x_i-y_{u_i})^2. \tag{11}$$

Accordingly, the optimization (6) can be rewritten as

$$\begin{aligned}
&\inf_{\mathcal{Y}}\min_{U(x)}\left[n\sum_{l=1}^{L}\left[-\frac{n_l}{n}\log\frac{n_l}{n}\right]+\lambda\sum_{i=1}^{n}(x_i-y_{u_i})^2\right]\\
=\;&\inf_{\mathcal{Y}}\min_{U(x)}\left[n\min_{q}\sum_{l=1}^{L}\left[-\frac{n_l}{n}\log q_l\right]+\lambda\sum_{i=1}^{n}(x_i-y_{u_i})^2\right]\\
=\;&\inf_{\mathcal{Y},q}\min_{U(x)}\left[n\sum_{l=1}^{L}\left[-\frac{n_l}{n}\log q_l\right]+\lambda\sum_{i=1}^{n}(x_i-y_{u_i})^2\right]\\
=\;&\inf_{\mathcal{Y},q}\min_{U(x)}\sum_{i=1}^{n}[-\log q_{u_i}+\lambda(x_i-y_{u_i})^2] \qquad (12)
\end{aligned}$$

where q is a pmf over $\mathcal{L}$. In comparison with the Lloyd algorithm for designing optimal variable rate scalar quantizers, it is easy to see that the optimization (12) is equivalent to (3.82) when $p(x)$ is the empirical distribution of $x = x_1 x_2 \cdots x_n$. Therefore, the Lloyd algorithm applied to discrete distributions can be used to solve (12). In particular, given $\mathcal{Y}$ and q, the optimal solution to the inner minimization in (12) is given by

$$u_i = \arg\min_{1 \le j \le L} [-\log q_j + \lambda (x_i - y_j)^2] \tag{13}$$

for $i = 1, 2, \cdots, n$. In this case, optimal SDQ is actually HDQ. On the other hand, given $U(x)$, the optimal solution to the outer minimization in (12) is given by

$$y_j = \frac{\sum_{i: u_i = j} x_i}{|\{i : u_i = j\}|} \tag{14}$$

for $j = 1, 2, \cdots, L$, and setting q to be the empirical distribution of $u(x)$.

Exercise: Hamming distortion and run length coding

Let $\mathcal{X} = \hat{\mathcal{X}} = \{0, 1\}$. Let d be the Hamming distortion measure. In this case, $Q_1(x) = \hat{\mathcal{X}}^n$ for all $x \in \mathcal{X}^n$. The optimization (6) is then reduced to

$$\inf_{\phi} \min_{U(x)} \left[\frac{|e(\phi)| + |\phi(U(x))|}{n} + \lambda d(x, U(x)) \right] \tag{15}$$

where $U(x) = Q_2(x) \in \hat{\mathcal{X}}^n$. Assume that ϕ is a run length coding algorithm. Solve the following minimization problem:

$$\min_{U(x)} \left[\frac{|\phi(U(x))|}{n} + \lambda d(x, U(x)) \right] \tag{16}$$

Then compare the resulting performance with $R(D)$ of an IID binary source when $x = x_1 x_2 \cdots x_n$ is drawn from the IID binary source.

Outline

As economic analysts at Monty Python LLC once describe corporate life in difficult times such as in recession,

> *There is nothing quite as wonderful as money.*
> *There is nothing quite as beautiful as cash.*

In comparison with efforts spent in industry to find better image/video coding algorithms by trial and error,

> *There is nothing quite as wonderful as the computational approach to lossy compression.*
> *There is nothing quite as beautiful as optimal SDQ.*

Fixed Product Reproduction Spaces

Fix $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\} \subset \hat{\mathcal{X}}$. Let $Q_1(x) = \mathcal{Y}^n$ for all $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$. Then the optimization (6) is reduced, in this case, to

$$\begin{aligned} &\inf_{\phi} \min_{U(x)} \left[\frac{|e(\phi)| + |\phi(U(x))|}{n} + \lambda d(x, Q_3(Q_1(x), U(x))) \right] \\ &= \min_{U(x)} \inf_{\phi} \left[\frac{|e(\phi)| + |\phi(U(x))|}{n} + \lambda d(x, Q_3(Q_1(x), U(x))) \right] \\ &= \min_{U(x)} \left[\frac{\inf_{\phi}[|e(\phi)\phi(U(x))|]}{n} + \lambda d(x, Q_3(Q_1(x), U(x))) \right]. \end{aligned} \tag{17}$$

Assume that the decoder of each ϕ can be implemented by a Turing machine, which is the case in practice. From the definition of Kolmogorov complexity, it then follows that (17) is equivalent to

$$\min_{U(x)} \left[\frac{K(U(x))}{n} + \lambda d(x, Q_3(Q_1(x), U(x))) \right] \tag{18}$$

where $K(U(x))$ is the Kolmogorov complexity of $U(x) \in \mathcal{L}^n$. Given the reproduction space $\mathcal{Y}^n$, (18) then gives the best rate distortion performance of each individual sequence $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$.

By turning (18) into constrained optimization, we get the notion of distortion Kolmogorov complexity defined first by Yang and Shen in 1993.

Distortion Kolmogorov complexity with respect to a fixed product reproduction space

For each $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$, its distortion Kolmogorov complexity with respect to $\mathcal{Y}^n$ is defined as

$$K_{\mathcal{Y}}(x, D) \stackrel{\Delta}{=} \min\{K(U(x)) : d(x, Q_3(Q_1(x), U(x))) \leq D\}. \tag{19}$$

Theorem (2 Equivalence between $K_{\mathcal{Y}}(x, D)$ and $R_{\mathcal{Y}}(D)$)

Let $X = \{X_i\}_{i=1}^{\infty}$ be a stationary ergodic source. Let $R_{\mathcal{Y}}(D)$ be the rate distortion function of X with respect to $\mathcal{Y}$ and d. Then for any $D > \inf\{D : R_{\mathcal{Y}}(D) < \infty\}$,

$$\frac{K_{\mathcal{Y}}(X_1 X_2 \cdots X_n, D)}{n} \to R_{\mathcal{Y}}(D) \tag{20}$$

with probability one as $n \to \infty$, i.e.,

$$\Pr\left\{\lim_{n\to\infty} \frac{K_{\mathcal{Y}}(X_1 X_2 \cdots X_n, D)}{n} = R_{\mathcal{Y}}(D)\right\} = 1.$$

Remark

Theorem 2 is the lossy counterpart of the equivalence result between Kolmogorov complexity and Shannon entropy when the source to be compressed is random and stationary.

Doubly universal codebooks

Fix $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\} \subset \hat{\mathcal{X}}$, and $R > 0$. For any $n \geq 1$, define

$$\mathbf{C}_n \stackrel{\Delta}{=} \{y_{u_1} y_{u_2} \cdots y_{u_n} : K(u_1 \cdots u_n) \leq nR - 1\}. \tag{21}$$

Then

$$|\mathbf{C}_n| \leq 2^{\lfloor nR \rfloor} \tag{22}$$

and $\mathbf{C}_n$, $n = 1, 2, \cdots$, give rise to a fixed rate lossy code $\mathcal{C} = (Q, \phi, g)$, where for any $x = x_1 x_2 \cdots x_n \in \mathcal{X}^n$,

$$Q_1(x) = \mathbf{C}_n, \;\; Q_2(x) = \arg\min_{C \in \mathbf{C}_n} d(x, C) \tag{23}$$

and ϕ encodes each $Q_2(x)$ by using a fixed number of bits $\lfloor nR \rfloor + 2\lfloor \log(n+1) \rfloor$.

Theorem (3 Universality)

Let $\mathcal{C} = (Q, \phi, g)$ be the fixed rate lossy code as defined in (23). Then for any stationary, ergodic source $X = \{X_i\}_{i=1}^{\infty}$,

$$d(X^n, Q_2(X^n)) \to D_{\mathcal{Y}}(R) \tag{24}$$

with probability one as $n \to \infty$, where $X^n = X_1 X_2 \cdots X_n$, and $D_{\mathcal{Y}}(R)$ is the unique distortion D such that $R_{\mathcal{Y}}(D) = R$.

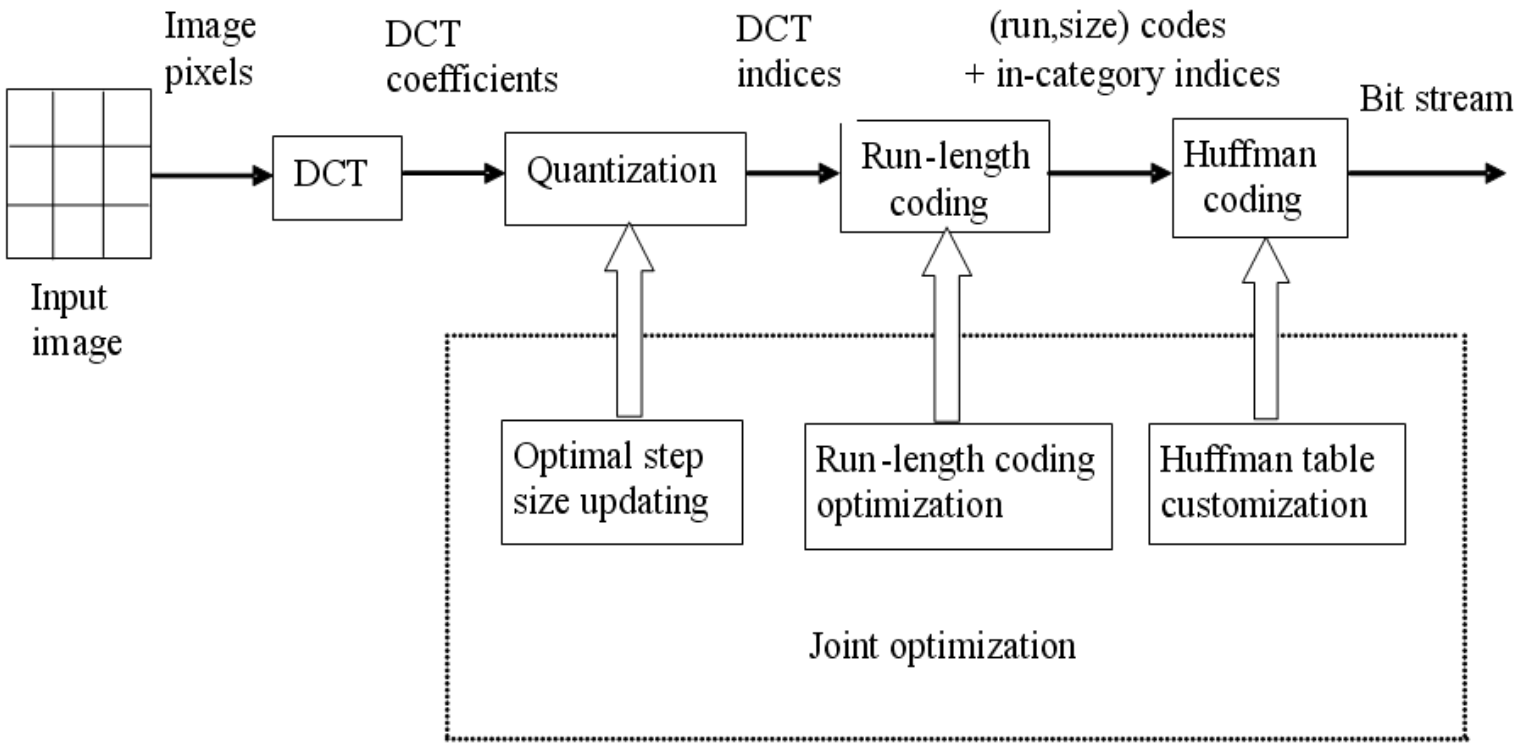
Remark

- Theorems 2 and 3 remain valid even if the Kolmogorov complexity $K(U(x))$ of $U(x)$ is replaced by the codeword length of $U(x)$ assigned by a universal lossless coding algorithm such as the Lempel-Ziv algorithms and Yang-Kieffer (grammar-based) coding algorithms.

- The construction of $\mathbf{C}_n$ in (21) is particularly interesting. It does not depend on the source to be compressed; it does not depend on the actual distortion measure either! In this sense, the codebooks $\mathbf{C}_n$, $n \geq 1$, are doubly universal.
- The key implication of these results is that lossless coding plays a vital role in lossy coding. If the lossless coding algorithm used in SDQ is efficient on its own, the resulting optimal SDQ can achieve (asymptotically) the theoretic rate distortion performance with respect to the given reproduction space. As such, the lossless coding algorithm used in SDQ should be designed in such a way that it is efficient on its own and at the same time it also facilitates the design of algorithms for optimal SDQ.

Outline

Introduction



Formal problem definition

Constrained optimization:

$$\min_{(R,S,ID),H,Q} d[I_0,(R,S,ID)_Q] \quad \text{subject to} \quad r[(R,S),H] \le r_{budget}$$

Equivalently:

$$\min_{(R,S,ID),H,Q} r[(R,S),H] \quad \text{subject to} \quad d[I_0,(R,S,ID)_Q] \le d_{budget}$$

Unconstrained optimization:

$$\min_{(R,S,ID),H,Q} \{J(\lambda) = d[I_0,(R,S,ID)_Q] + \lambda \cdot r[(R,S),H]\}$$

Problem solutions – iterative algorithm

1) Initialize a run-size distribution P_0 from the given image I_0 and a quantization table Q_0. Set $t = 0$, and specify a tolerance ε as the convergence criterion.
2) Fix P_t and Q_t for any $t \geq 0$. Find an optimal sequence (R_t, S_t, ID_t) that achieves the following minimum

$$\min_{(R,S,ID)} \{J(\lambda) = d[I_0, (R,S,ID)_{Q_t}] + \lambda \cdot r[(R,S), P_t]\}.$$

Denote $d[I_0, (R_t, S_t, ID_t)_{Q_t}] + \lambda \cdot r[(R_t, S_t), P_t]$ by $J^t(\lambda)$.

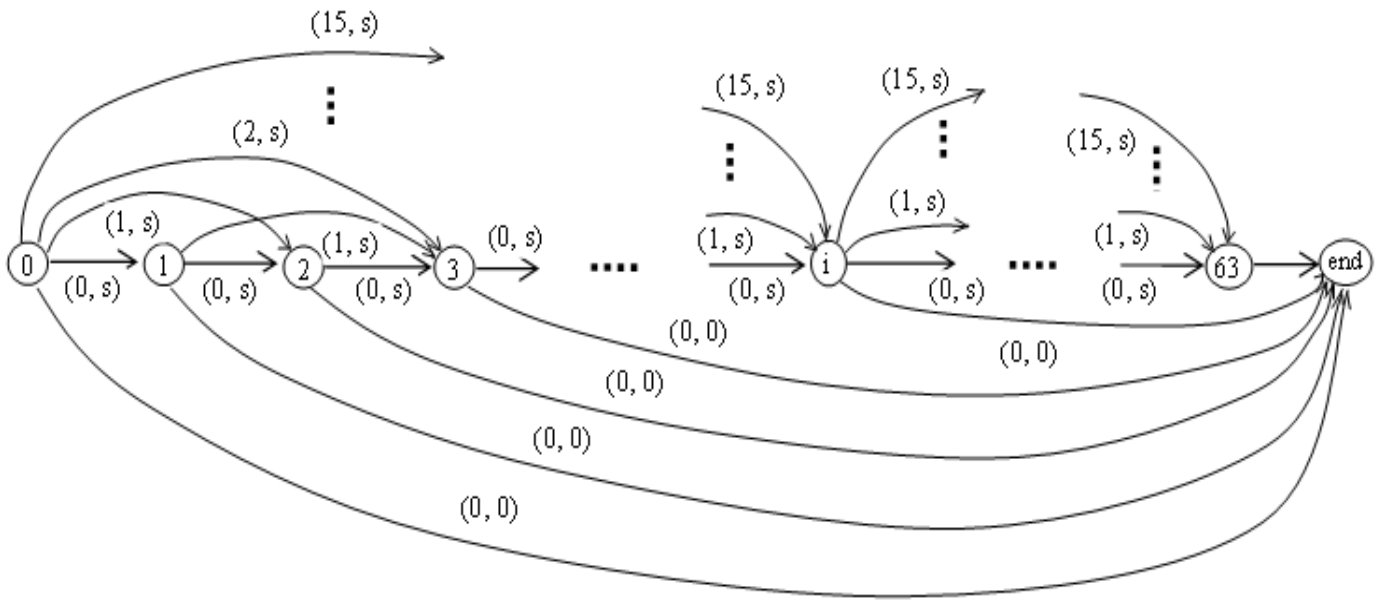
3) Fix (R_t, S_t, ID_t). Update Q_t and P_t into Q_{t+1} and P_{t+1}, respectively so that Q_{t+1} and P_{t+1} together achieve the following minimum

$$\min_{Q,P} \{J(\lambda) = d[I_0, (R_t, S_t, ID_t)_Q] + \lambda \cdot r[(R_t, S_t), P]\}$$

where the above minimization is taken over all quantization tables Q and all run-size probability distributions P. Note that P_{t+1} can be selected as the empirical run-size distribution of (R_t, S_t).

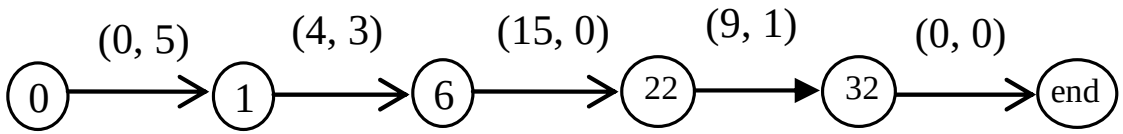
4) Repeat Steps 2) and 3) for $t = 0,1,2,\cdots$ until $J^t(\lambda) - J^{t+1}(\lambda) \leq \varepsilon$. Then output $(R_{t+1}, S_{t+1}, ID_{t+1})$, Q_{t+1} and P_{t+1}.

Problem solutions – graph-based run-length coding optimization



$$\sum_{j=i-r}^{i-1} C_j^2 + \left| C_i - q_i \cdot ID_i \right|^2 + \lambda \cdot (-\log_2 P(r,s) + s)$$

Problem solutions – graph-based run-length coding optimization: example



One-to-one mapping between a legitimate path from state 0 to the *end* state and a sequence of run-size pairs of an 8x8 block.

Problem solutions – optimal quantization table updating

$$\min_{Q} d[I_0,(R,S,ID)_Q]$$

$$d[I_0,(R,S,ID)_Q]=\sum_{i=1}^{63}\sum_{j=1}^{Num\_Blk}(C_{i,j}-q_i\cdot K_{i,j})^2$$

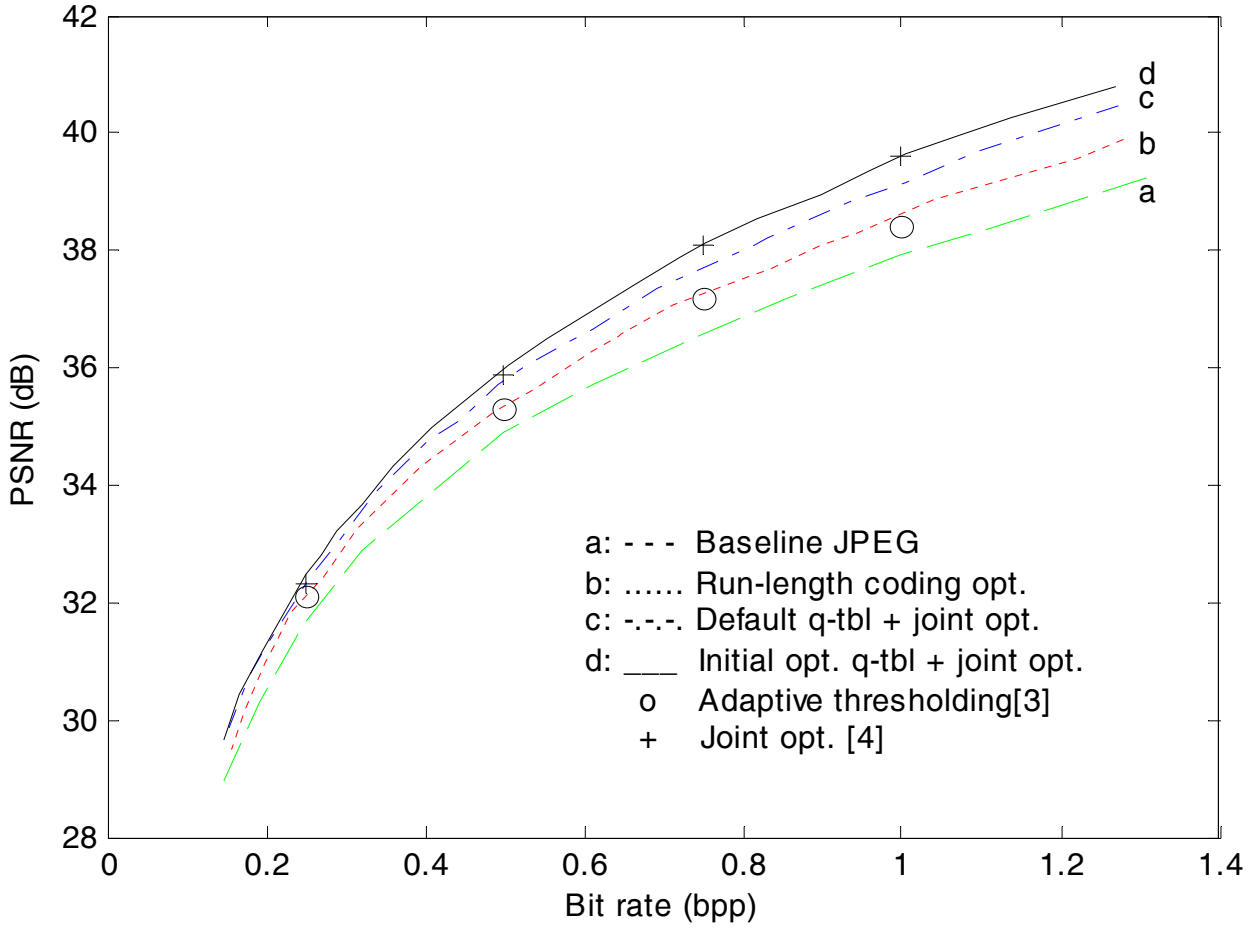
$$\min_{\hat{q}_i}\sum_{j=1}^{Num\_Blk}(C_{i,j}-\hat{q}_i\cdot K_{i,j})^2 \qquad i=1,\cdots,63$$

$$\hat{q}_i=\frac{\sum_{j=1}^{Num\_Blk}C_{i,j}\cdot K_{i,j}}{\sum_{j=1}^{Num\_Blk}K_{i,j}^{\,2}} \qquad i=1,\cdots,63$$

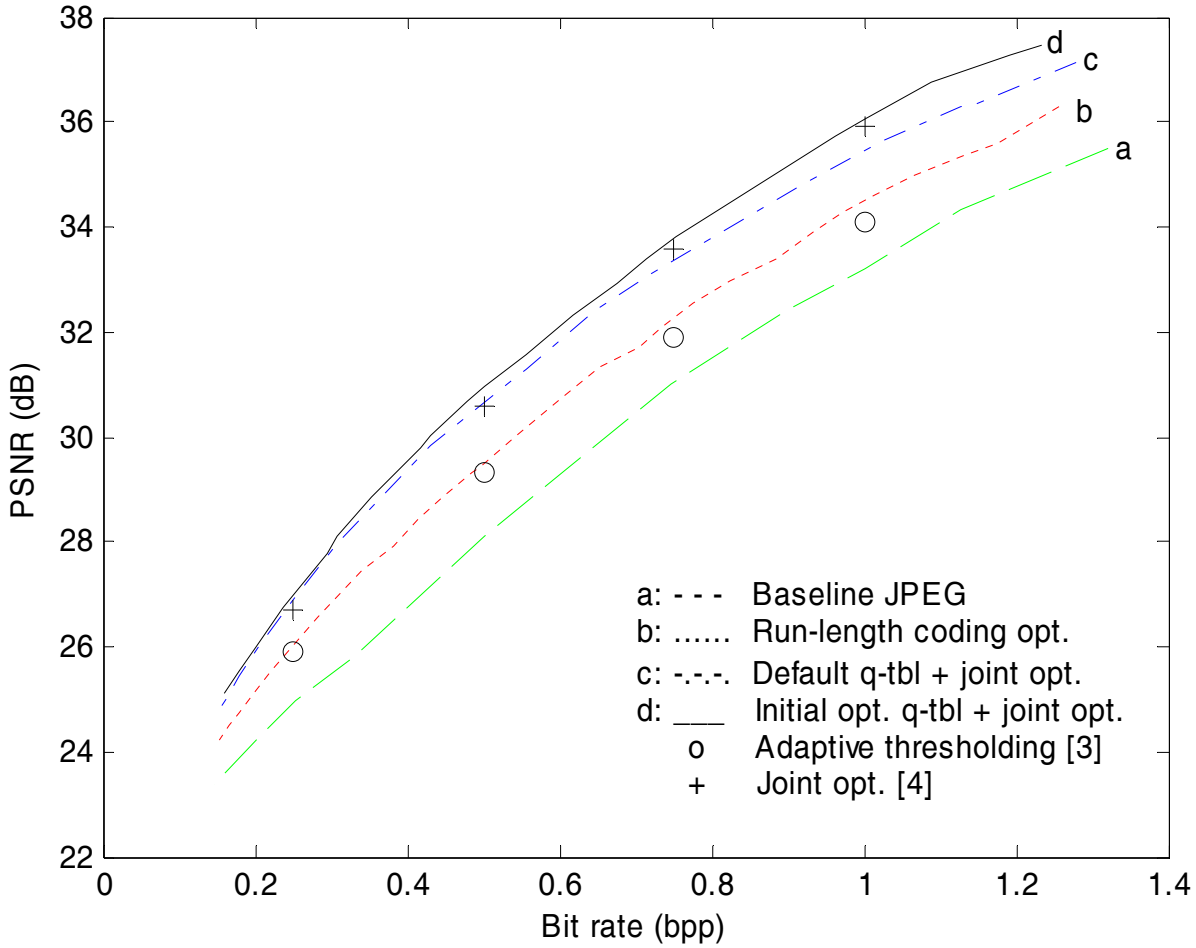
Experimental results – PSNR values

Image	Rate (bpp)	Customized baseline JPEG	Adaptive threshold [3]	Proposed run-length coding opt.	Default q-tbl + proposed joint opt.	Initially optimized q-tbl + proposed joint opt.	Joint opt. [4]	Baseline wavelet transform coder [7]	Embedded zerotree wavelet algorithm [8]
Lena	.25	31.63	32.1	32.21	32.37	**32.47**	32.3	33.17	33.17
	.50	34.90	35.3	35.43	35.80	**36.04**	35.9	36.18	36.28
	.75	36.62	37.2	37.32	37.68	**38.14**	38.1	38.02	N/A
	1.00	37.91	38.4	38.68	39.26	**39.63**	39.6	39.42	39.55
Barbara	.25	25.31	25.9	26.09	26.93	**27.04**	26.7	26.64	26.77
	.50	28.34	29.3	29.62	30.66	**30.94**	30.6	29.54	30.53
	.75	31.02	31.9	32.30	33.14	**33.82**	33.6	32.55	N/A
	1.00	33.16	34.1	34.52	35.23	**36.07**	35.9	34.56	35.14

Experimental results – R-D plot (Lena)



Experimental results – R-D plot (Barbara)



Experimental results – complexity glance

CPU time of the proposed joint optimization algorithm with one iteration on a Pentium PC (512x512 Lena)

Settings	Float DCT	Fast integer DCT
Comparing 3 size groups	**1.5 s**	**0.3 s**
Comparing 10 size groups	**2.0 s**	**0.7 s**