

Types of Coding

- Source Coding - Code data to more efficiently represent the information
 - Reduces “size” of data
 - Analog - Encode analog source data into a binary format
 - Digital - Reduce the “size” of digital source data
- Channel Coding - Code data for transmission over a noisy communication channel
 - Increases “size” of data
 - Digital - add redundancy to identify and correct errors
 - Analog - represent digital values by analog signals
- Complete “Information Theory” was developed by Claude Shannon

Digital Image Coding

- Images from a 6 MPixel digital cammera are 18 MBytes each
- Input and output images are digital
- Output image must be smaller (i.e. ≈ 500 kBytes)
- This is a digital source coding problem

Two Types of Source (Image) Coding

- Lossless coding (entropy coding)
 - Data can be decoded to form exactly the same bits
 - Used in “zip”
 - Can only achieve moderate compression (e.g. 2:1 - 3:1) for natural images
 - Can be important in certain applications such as medical imaging
- Lossless source coding
 - Decompressed image is visually similar, but has been changed
 - Used in “JPEG” and “MPEG”
 - Can achieve much greater compression (e.g. 20:1 - 40:1) for natural images
 - Uses entropy coding

Entropy

- Let X be a random variables taking values in the set $\{0, \dots, M - 1\}$ such that

$$p_i = P\{X = i\}$$

- Then we define the entropy of X as

$$\begin{aligned} H(X) &= - \sum_{i=0}^{M-1} p_i \log_2 p_i \\ &= -E [\log_2 p_X] \end{aligned}$$

$H(X)$ has units of bits

Conditional Entropy and Mutual Information

- Let (X, Y) be a random variables taking values in the set $\{0, \dots, M - 1\}^2$ such that

$$p(i, j) = P\{X = i, Y = j\}$$

$$p(i|j) = \frac{p(i, j)}{\sum_{k=0}^{M-1} p(k, j)}$$

- Then we define the conditional entropy of X given Y as

$$\begin{aligned} H(X|Y) &= - \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} p(i, j) \log_2 p(i|j) \\ &= -E [\log_2 p(X|Y)] \end{aligned}$$

- The mutual information between X and Y is given by

$$I(X; Y) = H(X) - H(X|Y)$$

The mutual information is the reduction in uncertainty of X given Y .

Entropy (Lossless) Coding of a Sequence

- Let X_n be an i.i.d. sequence of random variables taking values in the set $\{0, \dots, M - 1\}$ such that

$$P\{X_n = m\} = p_m$$

– X_n for each n is known as a symbol

- How do we represent X_n with a minimum number of bits per symbol?

A Code

- **Definition:** A code is a mapping from the discrete set of symbols $\{0, \dots, M-1\}$ to finite binary sequences
 - For each symbol, m there is a corresponding finite binary sequence σ_m
 - $|\sigma_m|$ is the length of the binary sequence
- Expected number of bits per symbol (bit rate)

$$\begin{aligned}\bar{n} &= E[|\sigma_{X_n}|] \\ &= \sum_{m=0}^{M-1} |\sigma_m| p_m\end{aligned}$$

- Example for $M = 4$

m	\rightarrow	σ_m	$ \sigma_m $
0	\rightarrow	01	2
1	\rightarrow	10	2
2	\rightarrow	0	1
3	\rightarrow	100100	6

- Encoded bit stream

$$(0, 2, 1, 3, 2) \rightarrow (01|0|10|100100|0)$$

Fixed versus Variable Length Codes

- Fixed Length Code - $|\sigma_m|$ is constant for all m
- Variable Length Code - $|\sigma_m|$ varies with m
- Problem
 - Variable length codes may not be uniquely decodable
 - Example: Using code from previous page

$$(6) \rightarrow (100100)$$

$$(1, 0, 2, 2) \rightarrow (10|01|0|0)$$

- Different symbol sequences can yield the same code
- **Definition:** A code is *Uniquely Decodable* if there exists only a single unique decoding of each coded sequence.
- **Definition:** A *Prefix Code* is a specific type of uniquely decodable code in which no code is a prefix of another code.

Lower Bound on Bit Rate

- **Theorem:** Let C be a uniquely decodable code for the i.i.d. symbol sequence X_n . Then the mean code length is greater than $H(X_n)$.

$$\begin{aligned}\bar{n} &\triangleq e[|\sigma_{X_n}|] \\ &= \sum_{m=0}^{M-1} |\sigma_m| p_m \\ &\geq H(X_n)\end{aligned}$$

- Question: Can we achieve this bound?
- Answer: Yes! Constructive proof using Huffman codes

Huffman Codes

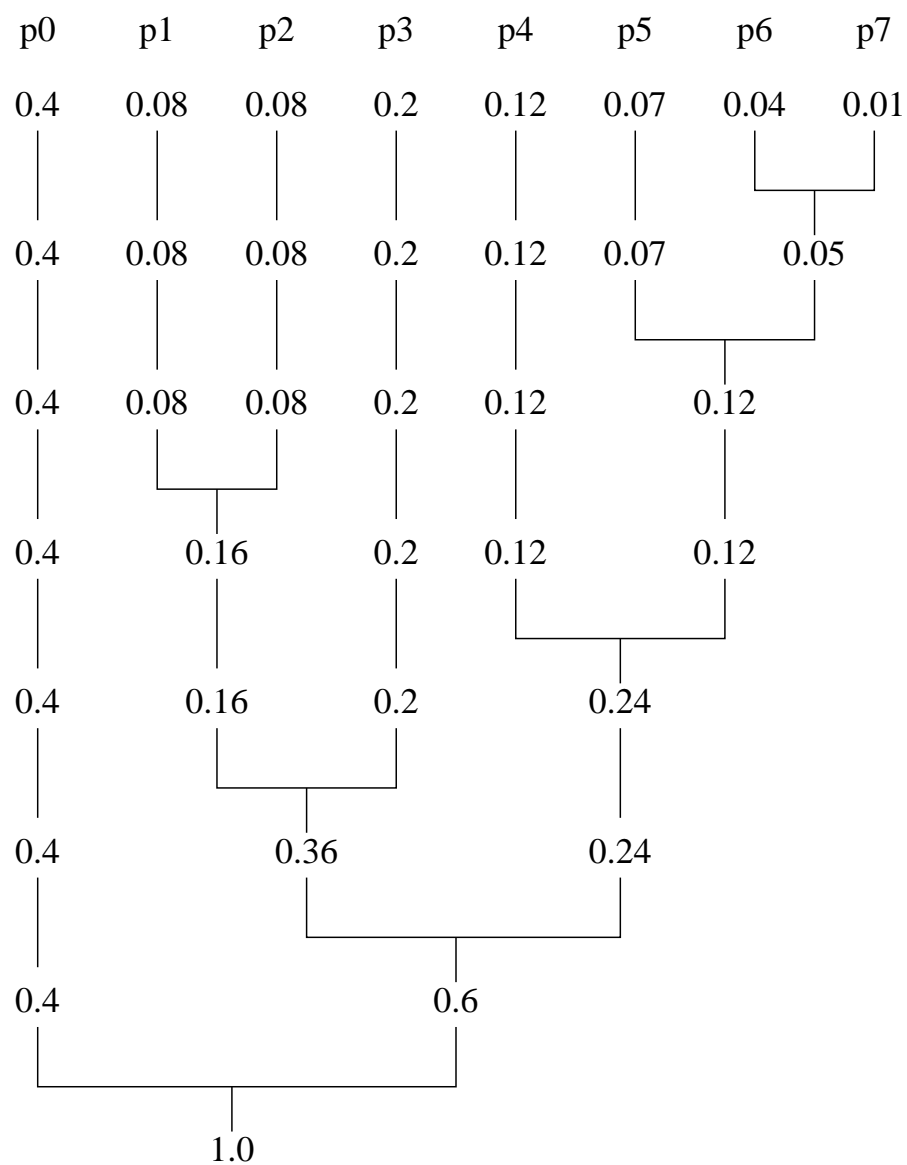
- Variable length prefix code \Rightarrow Uniquely decodable
- Basic idea:
 - Low probability symbols \Rightarrow Long codes
 - High probability symbols \Rightarrow short codes
- Basic algorithm:
 - Low probability symbols \Rightarrow Long codes
 - High probability symbols \Rightarrow short codes

Huffman Coding Algorithm

1. Initialize list of probabilities with the probability of each symbol
2. Search list of probabilities for two smallest probabilities, p_{k*} and p_{l*} .
3. Add two smallest probabilities to form a new probability, $p_m = p_{k*} + p_{l*}$.
4. Remove p_{k*} and p_{l*} from the list.
5. Add p_m to the list.
6. Go to step 2 until the list only contains 1 entry

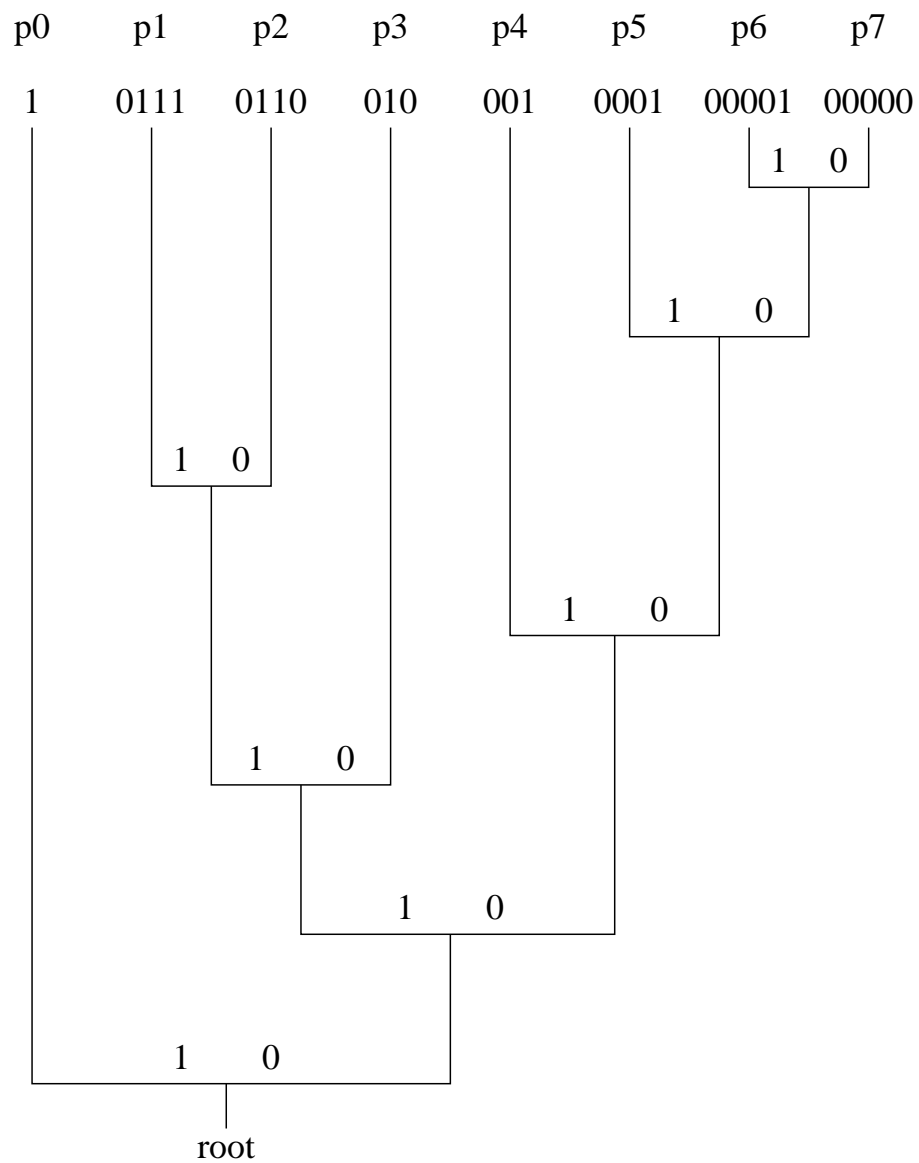
Recursive Merging for Huffman Code

- Example for $M = 8$ code



Resulting Huffman Code

- Binary codes given by path through tree



Upper Bound on Bit Rate of Huffman Code

- **Theorem:** For a Huffman code, \bar{n} has the property that

$$H(X_n) \leq \bar{n} < H(X_n) + 1$$

- A Huffman code is within 1 bit of optimal efficiency
- Can we do better?

Coding in Blocks

- We can code blocks of symbols to achieve a bit rate that approaches the entropy of the source symbols.

$$\cdots, \underbrace{X_0, \cdots, X_{m-1}}_{Y_0}, \underbrace{X_m, \cdots, X_{2m-1}}_{Y_1}, \cdots$$

So we have that

$$Y_n = [X_{nm}, \cdots, X_{(n+1)m-1}]$$

where

$$Y_n \in \{0, \cdots, M^m - 1\}$$

Bit Rate Bounds for Coding in Blocks

- It is easily shown that $H(Y_n) = mH(X_n)$ and the number of bits per symbol X_n is given by $\bar{n}_x = \frac{\bar{n}_y}{m}$ where \bar{n}_y is the number of bits per symbol for a Huffman code of Y_n .
- Then we have that

$$H(Y_n) \leq \bar{n}_y < H(Y_n) + 1$$

$$\frac{1}{m}H(Y_n) \leq \frac{\bar{n}_y}{m} < \frac{1}{m}H(Y_n) + \frac{1}{m}$$

$$H(X_n) \leq \frac{\bar{n}_y}{m} < H(X_n) + \frac{1}{m}$$

$$H(X_n) \leq \bar{n}_x < H(X_n) + \frac{1}{m}$$

- As the block size grows, we have

$$\lim_{m \rightarrow \infty} H(X_n) \leq \lim_{m \rightarrow \infty} \bar{n}_x \leq H(X_n) + \lim_{m \rightarrow \infty} \frac{1}{m}$$

$$H(X_n) \leq \lim_{m \rightarrow \infty} \bar{n}_x \leq H(X_n)$$

- So we see that for a Huffman code of blocks with length m

$$\lim_{m \rightarrow \infty} \bar{n}_x = H(X_n)$$

Comments on Entropy Coding

- As the block size goes to infinity the bit rate approaches the entropy of the source

$$\lim_{m \rightarrow \infty} \bar{n}_x = H(X_n)$$

- A Huffman coder can achieve this performance, but it requires a large block size.
- As m becomes large M^m becomes very large \Rightarrow large blocks are not practical.
- This assumes that X_n are i.i.d., but a similar result holds for stationary and ergodic sources.
- Arithmetic coders can be used to achieve this bitrate in practical situations.

Run Length Coding

- In some cases, long runs of symbols may occur. In this case, run length coding can be effect as a preprocessor to an entropy coder.
- Typical run length coder uses
 $\dots, (\text{value}, \# \text{ of repetitions}), (\text{value}, \# \text{ of repetitions}+1), \dots$

where 2^b is the maximum number of repetitions

- Example: Let $X_n \in \{0, 1, 2\}$

$$\dots | \underbrace{00000000}_{07} | \underbrace{111}_{13} | \underbrace{222222}_{26} | \dots$$

- If more than 2^b repetitions occur, then the repetition is broken into segments

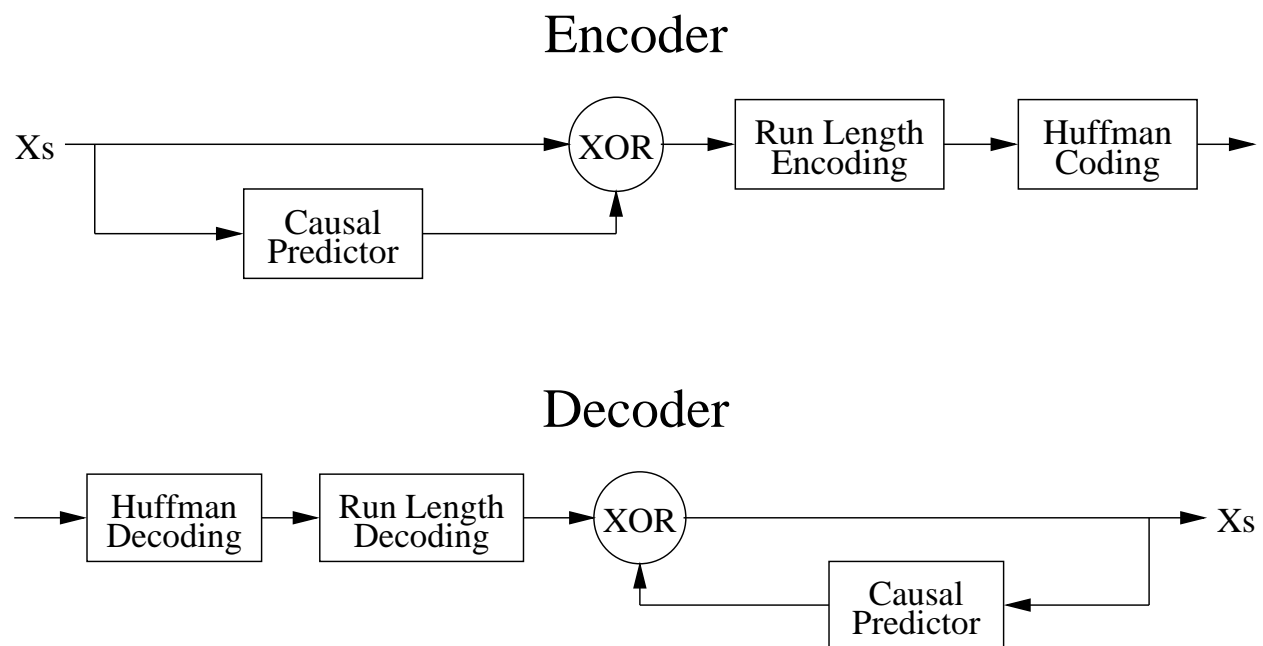
$$\dots | \underbrace{00000000}_{08} | \underbrace{00}_{02} | \underbrace{111}_{13} | \dots$$

- Many other variations are possible.

Predictive Entropy Coder for Binary Images

- Uses in transmission of Fax images (CCITT G4 standard)
- Framework
 - Let X_s be a binary image on a rectangular lattice
 $s = (s_1, s_2) \in S$
 - Let W be a causal window in raster order
 - Determine a model for $p(x_s | x_{s+r} r \in W)$
- Algorithm
 1. For each pixel in raster order
 - (a) Predict
$$\hat{X}_s = \begin{cases} 1 & \text{if } p(1 | X_{s+r} r \in W) > p(0 | X_{s+r} r \in W) \\ 0 & \text{otherwise} \end{cases}$$
 - (b) If $X_s = \hat{X}_s$ send 0; otherwise send 1
 2. Run length code the result
 3. Entropy code the result

Predictive Entropy Coder Flow Diagram



How to Choose $p(x_s | x_{s+r} \ r \in W)$?

- Non-adaptive method
 - Select typical set of training images
 - Design predictor based on training images
- Adaptive method
 - Allow predictor to adapt to images being coded
 - Design decoder so it adapts in same manner as encoder

Non-Adaptive Predictive Coder

- Method for estimating predictor
 1. Select typical set of training images
 2. For each pixel in each image, form $z_s = (x_{s+r_0}, \dots, x_{s+r_{p-1}})$ where $\{r_0, \dots, r_{p-1}\} \in W$.
 3. Index the values of z_s from $j = 0$ to $j = 2^p - 1$
 4. For each pixel in each image, compute

$$h_s(i, j) = \delta(x_s = i) \delta(z_s = j)$$

and the histogram

$$h(i, j) = \sum_{s \in S} h_s(i, j)$$

5. Estimate $p(x_s | x_{s+r} \ r \in W) = p(x_s | z_s)$ as

$$\hat{p}(x_s = i | z_s = j) = \frac{h(i, j)}{\sum_{k=0}^1 h(k, j)}$$

Adaptive Predictive Coder

- Adapt predictor at each pixel
- Update value of $h(i, j)$ at each pixel using equations

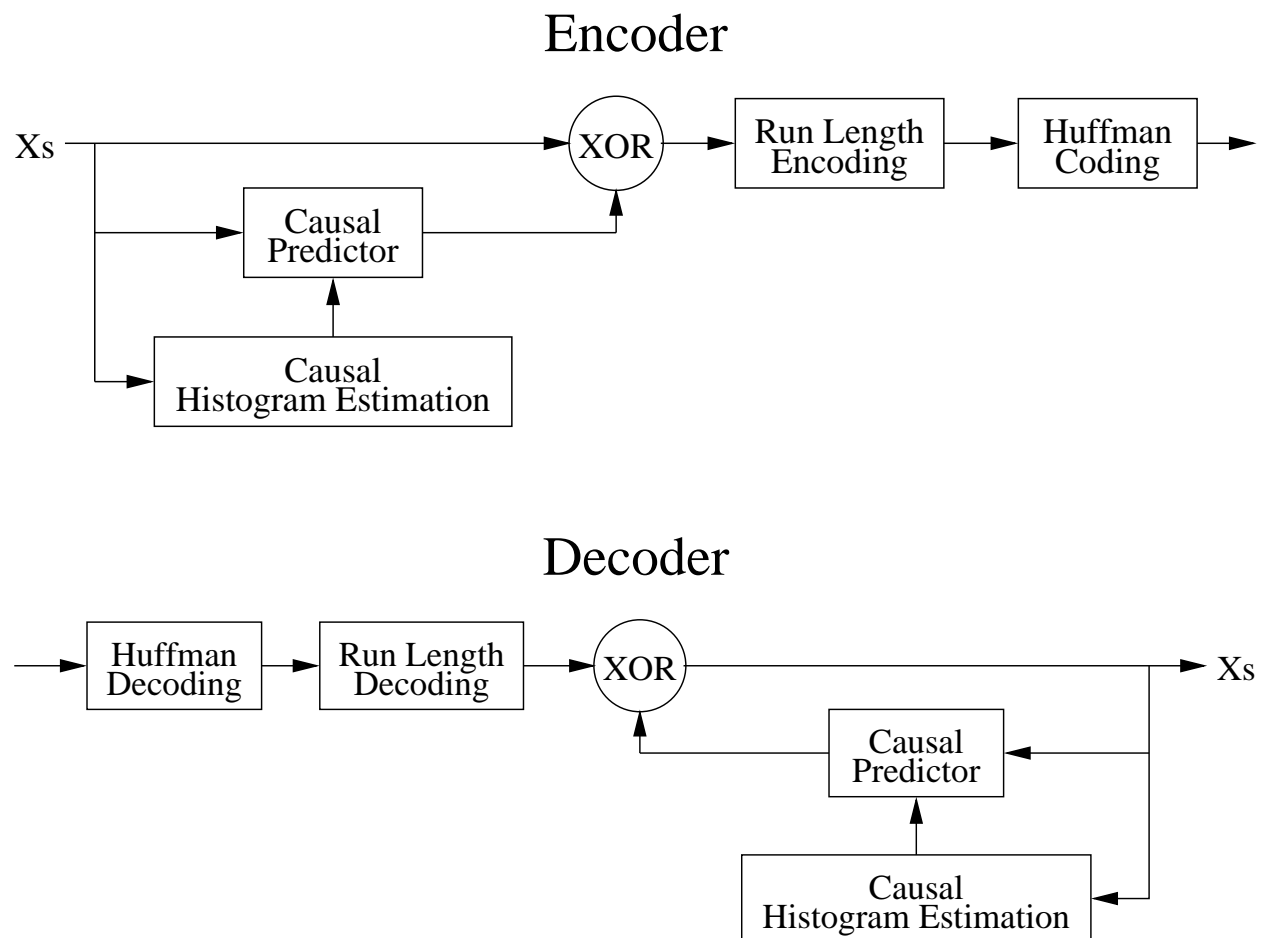
$$h(i, j) \leftarrow h(i, j) + \delta(x_s = i)\delta(z_s = j)$$

$$N \leftarrow N + 1$$

$$h(i, j) \leftarrow \frac{h(i, j)}{N}$$

- Use updated values of $h(i, j)$ to compute new predictor at each pixel
- Design decoder to track encoder

Adaptive Predictive Entropy Coder Flow Diagram



Lossy Source Coding

- Method for representing discrete-space signals with minimum distortion and bit-rate
- Rate-distortion theory
- Karhunen-Loeve decorrelating Transform
- Practical coder structures

Distortion

- Let X and Z be random vectors in \mathbb{R}^M . Intuitively, X is the original image/data and Z is the decoded image/data.

Assume we use the squared error distortion measure given by

$$d(X, Y) = \|X - Z\|^2$$

Then the distortion is given by

$$D = E[d(X, Y)] = E[\|X - Z\|^2]$$

- This actually applies to any quadratic norm error distortion measure since we can define

$$\tilde{X} = AX \quad \text{and} \quad \tilde{Z} = AZ$$

So

$$\tilde{D} = E[\|\tilde{X} - \tilde{Z}\|^2] = E[\|X - Z\|_B^2]$$

where $B = A^t A$.

Lossy Source Coding

- Notation for source coding

X_0, \dots, X_{N-1} - a sequence of N i.i.d. random vectors each with distribution the same as X .

$Y \in \{0, 1\}^K$ - a K bit random binary vector.

(Z_0, \dots, Z_{N-1}) - the decoded sequence of N random vectors.

– Encoder function: $Y = Q(X_0, \dots, X_{N-1})$

– Decoder function: $(Z_0, \dots, Z_{N-1}) = f(Y)$

- Resulting quantities

$$\text{Bit-rate} = \frac{K}{N}$$

$$\text{Distortion} = D(X, Z) = \frac{1}{N} \sum_{n=0}^{N-1} E \left[\|X_n - Z_n\|^2 \right]$$

- How do we choose Y to minimize the bit-rate and distortion?

Differential Entropy

- Notice that the information contained in a Gaussian random variable is infinite, so the conventional entropy $H(X)$ is not defined.
- Let X be a random vector taking values in \mathbb{R}^M with density function $p(x)$. Then we define the differential entropy of X as

$$\begin{aligned} h(X) &= - \int_{x \in \mathbb{R}^M} p(x) \log_2 p(x) dx \\ &= -E [\log_2 p(X)] \end{aligned}$$

$h(X)$ has units of bits

Conditional Entropy and Mutual Information

- Let X and Y be a random vectors taking values in \mathbb{R}^M with density function $p(x, y)$ and conditional density $p(x|y)$.
- Then we define the differential conditional entropy of X given Y as

$$\begin{aligned} h(X|Y) &= - \int_{x \in \mathbb{R}^M} \int_{y \in \mathbb{R}^M} p(x, y) \log_2 p(x|y) \\ &= -E [\log_2 p(X|Y)] \end{aligned}$$

- The mutual information between X and Y is given by

$$I(X; Y) = h(X) - h(X|Y) = I(Y; X)$$

- **Important:** The mutual information is well defined for both continuous and discrete random variables, and it represents the reduction in uncertainty of X given Y .

The Theoretical Rate

- We will define a theoretical quantity called “rate” which we will later show is the optimum bit-rate that can be achieved for a given distortion.
- For any $\delta > 0$, the rate is given by

$$R(\delta) = \inf_Z \{I(X; Z) : D \leq \delta\}$$

where the infimum (i.e. minimum) over Z is taken over all random variables Z .

- Properties of $R(\delta)$
 - $R(\delta)$ is a monotone decreasing function of δ .
 - If $\delta \geq E[||X||^2]$, then $R(\delta) = 0$
 - $R(\delta)$ is a convex function of δ

Shannon's Source-Coding Theorem

- Shannon's Source-Coding Theorem:

For any $R' > R(\delta)$ and $D' > \delta$ there exists a sufficiently large N such that there is an encoder

$$Y = Q(X_0, \dots, X_{N-1})$$

which achieves

$$\frac{K}{N} \leq R'$$

and

$$D(X, Z) \leq D'.$$

- Comments:

- One can achieve a bit rate arbitrarily close to $R(\delta)$ at a distortion $D(\delta)$.
- Proof is constructive (but not practical), and uses codes that are randomly distributed in the space \mathbb{R}^{MN} of source symbols.