Chapter 1: Source Coding

Gautam Singh

Contents

1 Conventions

1.1 X denotes a random variable.
1.2 $\mathcal{X}$ denotes the alphabet.
1.3 x denotes a particular value of the alphabet.
1.4 $p(x) \triangleq \Pr(X = x)$.

2 Uncertainity and Information

2.1 The **self information** of the event $X = x$ is defined as

$$I(x) \triangleq \log\left(\frac{1}{\log p(x)}\right) = -\log(p(x)) \tag{2.1}$$

Clearly, $I(x) = 0$ at $p(x) = 1$, that is, a high probability event conveys lesser information.

2.2 The units are determined by the base of the algorithm

2.2.1 If the base is 2, the units are **bits**.
2.2.2 If the base is e, the units are **nats**.
2.2.3 If the base is 10, the units are **dits**.

2.3 The **mutual information** between x and y is defined as

$$I(x;y) \triangleq \log\left(\frac{p(x|y)}{p(x)}\right) \tag{2.2}$$

Observe that

$$I(x;y) = \log\left(\frac{p(x|y)}{p(x)}\right) \tag{2.3}$$

$$= \log\left(\frac{p(x|y)p(y)}{p(x)p(y)}\right) \tag{2.4}$$

$$= \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2.5}$$

$$= \log\left(\frac{p(y|x)}{p(y)}\right) = I(y;x) \tag{2.6}$$

It can be interpreted as the information event $Y = y$ provides about $X = x$.

2.4 From (2.6), the amount of information about $X = x$ provided by $Y = y$ is the same as the amount of information about $Y = y$ provided by $X = x$.

2.5 Notice that when X and Y are independent, then $I(x;y) = 0$ as $p(x,y) = p(x)p(y)$. Similarly, if $p(x|y) = 1$, then $I(x;y) = I(x)$.

2.6 The **conditional self information** of the event $X = x$ given $Y = y$ is defined as

$$I(x|y) \triangleq \log\left(\frac{1}{p(x|y)}\right) = -\log p(x|y) \tag{2.7}$$

Notice that

$$I(x;y) = \log\left(\frac{p(x|y)}{p(x)}\right) = I(x) - I(x|y) \tag{2.8}$$

And thus mutual information can be positive, negative or zero.

3 Average Mutual Information and Entropy

3.1 The **average mutual information** between random variables X and Y is defined as

$$I(X;Y) \triangleq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)T(x;y) \tag{3.1}$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{3.2}$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log\left(\frac{p(y|x)}{p(x)}\right) \tag{3.3}$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(y)p(x|y)\log\left(\frac{p(x|y)}{p(y)}\right) \tag{3.4}$$

$$= E\left[\log\left(\frac{p(X,Y)}{p(X)p(Y)}\right)\right] \tag{3.5}$$

$$= E\left[-\log\left(\frac{p(X)p(Y)}{p(X,Y)}\right)\right] \tag{3.6}$$

3.2 When X and Y are independent, (3.5) gives $I(X;Y) = 0$, that is, there is no average information between X and Y.

3.3 In general, $I(X;Y) \geq 0$ with equality iff X and Y are independent.

3.4 The **average self information** or **entropy** of a random variable X is defined as

$$H(X) \triangleq \sum_{x\in\mathcal{X}} p(x)I(x) \tag{3.7}$$

$$= \sum_{x\in\mathcal{X}} p(x)\log\left(\frac{1}{p(x)}\right) \tag{3.8}$$

$$= E\left[\log\left(\frac{1}{p(X)}\right)\right] \tag{3.9}$$

$$= E\left[-\log p(X)\right] \tag{3.10}$$

3.5 Notice that since $0 \leq p(x) \leq 1$, (3.7) gives $H(X) \geq 0$.

3.6 The units of $I(X;Y)$ and $H(X)$ are **bits**.

3.7 For a Bernoulli trial with success rate p, the entropy of the outcome X is

$$H(X) = -\left(p\log_2 p + (1-p)\log_2(1-p)\right) \tag{3.11}$$

which is known as the *binary entropy function* and denoted by $h_2(p)$.

3.8 The **average self information** or **conditional entropy** of a random variable X given a random variable Y is defined as

$$H(X|Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{1}{p(x|y)} \tag{3.12}$$

$$= E\left[\log\frac{1}{p(X|Y)}\right] \tag{3.13}$$

$$= E\left[-\log p(X|Y)\right] \tag{3.14}$$

Clearly,

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3.15}$$

3.9 Note that $I(X;Y) \geq 0 \implies H(X) \geq H(X|Y)$. Thus, conditioning can only decrease entropy. In case it does not, X and Y are independent.

3.10 The **joint entropy** of a pair of discrete random variables (X,Y) with a joint pmf $p(x,y)$ is defined as

$$H(X,Y) \triangleq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{1}{p(x,y)} \tag{3.16}$$

$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x,y) \tag{3.17}$$

$$= E\left[\log\frac{1}{p(X,Y)}\right] \tag{3.18}$$

$$= E\left[-\log p(X,Y)\right] \tag{3.19}$$

In general, the joint entropy of an n-tuple of random variables $(X_1, X_2, \ldots, X_n)$ with joint pmf $p(X_1, X_2, \ldots, X_n)$ is

$$H(X_1, X_2, \ldots, X_n) \triangleq E\left[-\log p(X_1, X_2, \ldots, X_n)\right] \tag{3.20}$$

3.11 From (3.10) and (3.14), we get the **chain rule**

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) \tag{3.21}$$

In general for n random variables $X_i,\ 1 \leq i \leq n$, the chain rule is

$$H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_1, X_2, \ldots, X_{i-1}) \tag{3.22}$$

3.12 From (3.6), we clearly see

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{3.23}$$

4 Information Measures for Continuous Random Variables

4.1 The **average mutual information** between two continuous random variables X and Y with joint pdf $p(x,y)$ and marginal pdfs $p(x)$ and $p(y)$ respectively is defined as

$$I(X;Y) \triangleq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y|x)\log\frac{p(y|x)}{p(y)}\,dxdy \tag{4.1}$$

4.2 Note that while physical interpretation of mutual information can be applied here, such physical interpretations will not work with other quantities. This is because the information in a continuous random variable is infinite. Hence, differential entropy is defined.

4.3 The **differential entropy** of a continuous random variable X is defined as

$$h(X) \triangleq -\int_{-\infty}^{\infty} p(x)\log p(x)\,dx \tag{4.2}$$

While there is no physical meaning for this quantity, the units remain bits.

4.4 The **average conditional entropy** of a continuous random variable X given Y is defined as

$$h(X|Y) \triangleq -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log p(x|y)\,dxdy \tag{4.3}$$

We can express the *average mutual information* as

$$I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X) \tag{4.4}$$

4.5 Rules for differential entropy:

4.5.1 The chain rule for differential entropy is given as

$$h(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} h(X_i|X_1, X_2, \ldots, X_{i-1}) \tag{4.5}$$

4.5.2 $h(X+c) = h(X)$, that is, transition does not alter differential entropy.

4.5.3 $h(aX) = h(X) + \log|a|$.

4.5.4 If X and Y are independent, then we have $h(X+Y) \geq h(X+Y|Y) = h(X|Y) = h(X)$.

5 Relative Entropy

5.1 The **relative entropy** or **Kullback Leibler Distance** between two pmfs $p(x)$ and $q(x)$ is defined as

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right) \tag{5.1}$$

$$= E\left[\log\left(\frac{p(X)}{q(X)}\right)\right] \tag{5.2}$$

It is sometimes denoted by $D_{KL}(p||q)$.

5.2 We can rewrite the mutual information in terms of relative entropy

$$I(X;Y) = D(p(x,y)||p(x)p(y)) \tag{5.3}$$

5.3 The **Jensen Shannon distance** between two pmfs $p(x)$ and $q(x)$ is defined as

$$JSD(p||q) \triangleq \frac{1}{2}\left(D(p||m) + D(q||m)\right) \tag{5.4}$$

It is sometimes denoted by $D_{JS}(p||q)$, and referred to as **Jensen Shannon Divergence** or **Information Radius**.

5.4 A function f defined on $[0,1]$ is said to be **convex** if

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \tag{5.5}$$

for all $\lambda \in [0,1]$. On the other hand, if f is **concave**, then the inequality in (5.5) becomes $\geq$. Strict inequalities would make f **strictly convex** or **strictly concave**.

5.5 $D(p||q)$ is convex in the pair (p,q) and $H(p)$ is concave in p.

6 Source Coding Theorem

6.1 A **code** is a set of vectors called **codewords**.

6.2 A code where all codewords have the same length is called **fixed length code (FLC)**.

6.3 When source symbols are not equally probable, it is more efficient to use a **variable length code (VLC)**.

6.4 A **prefix code** is a code in which no codeword is a prefix of any other codeword. This condition is also called the **prefix condition**. Prefix codes are also called **instantaneous codes** since they can be decoded without any delay.

6.5 **Kraft Inequality:** A necessary and sufficient condition for the existence of a binary code

with codewords of lengths $n_1 \leq n_2 \leq \ldots \leq n_L$ satisfying the prefix condition is

$$\sum_{k=1}^{L} 2^{-n_k} \leq 1 \tag{6.1}$$

Proof. Consider a binary tree of depth n_L. For the necessary condition, note that selecting a codeword of length n_i eliminates $2^{n_L - n_i}$ terminal nodes, since the code is prefix-free. Thus, at the node of order j, the fraction of terminal nodes eliminated is

$$\sum_{i=1}^{j} 2^{-n_i} < \sum_{i=1}^{L} 2^{-n_i} \leq 1 \tag{6.2}$$

and we can create a prefix-free code with the given lengths.

To prove the sufficient condition, note that the total number of terminal nodes eliminated is upto 2^{n_L}. Thus,

$$\sum_{i=1}^{L} 2^{n_L - n_i} \leq 2^{n_L} \tag{6.3}$$

$$\implies \sum_{i=1}^{L} 2^{-n_i} \leq 1 \tag{6.4}$$

□

6.6 The Kraft inequality holds for M-ary code, where $M > 1$ is an integer.

6.7 **Source Coding Theorem:** It is possible to construct a prefix-free code that has an average length $\bar{L}$ satisfying

$$H(X) \leq \bar{L} < H(X) + 1 \tag{6.5}$$

Proof. Let the length of the codeword of symbol x be $l(x)$. Then,

$$H(X) - \bar{L} = \sum_{x \in \mathcal{X}} p(x) \log \frac{2^{-l(x)}}{p(x)} \tag{6.6}$$

$$\leq \log e \sum_{x \in \mathcal{X}} p(x) \left(\frac{2^{-l(x)}}{p(x)} - 1 \right) \tag{6.7}$$

$$= \log e \left(\sum_{x \in \mathcal{X}} 2^{-l(x)} - 1 \right) \leq 0 \tag{6.8}$$

We use the fact that $\ln x \leq x - 1$ in (6.6) and the Kraft inequality in (6.8). This proves the lower bound on $\bar{L}$.

To prove the upper bound, choose codewords for each symbol x satisfying

$$2^{-l(x)} \leq p(x) \leq 2^{-l(x)+1} \tag{6.9}$$

Consider the left hand side of (6.9). Summing over all symbols,

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq \sum_{x \in \mathcal{X}} p(x) = 1 \tag{6.10}$$

This satisfies the Kraft inequality, hence such a codeword can be selected. Now, considering the right hand side of (6.9), taking logarithms gives

$$\log p(x) < 1 - l(x) \tag{6.11}$$

$$\implies \log p(x) + l(x) < 1 \tag{6.12}$$

Thus, using (6.12),

$$\bar{L} - H(X) = \sum_{x \in \mathcal{X}} p(x) \left(l(x) + \log p(x) \right) \tag{6.13}$$

$$< \sum_{x \in \mathcal{X}} p(x) = 1 \tag{6.14}$$

which proves the upper bound. □

6.8 The source coding theorem tells us that the minimum number of bits needed to represent a discrete random variable is at least the entropy of the random variable.

6.9 The **efficiency** of a prefix-free code is defined as

$$\eta \triangleq \frac{H(X)}{\bar{L}} \tag{6.15}$$

Note that $\eta \leq 1$ by the source coding theorem.

7 Huffman Coding

7.1 Variable length encoding algorithm based on source symbol probabilities.

7.2 Steps of algorithm:

a) Arrange the source symbols in *decreasing* order of probabilities.

b) Take the smallest two probabilities, and tie them together. Add their probabilities and write it on the combined node. Label the two branches with a '1' and '0'.

c) Repeat this with the next two smallest probabilities, treating the computed probability as the probability of a new symbol, till one probability is left and it equals 1. This is the construction of a **Huffman Tree.**

d) To find the codeword, follow the branches from the final node back to the symbol.
e) **Note:** We can have more than one possible Huffman coding scheme.

7.3 Optimality is achieved when the entropy of each symbol is an integer. Thus, the probabilities of each symbol is a negative power of 2.

7.4 **Note:** A D-adic distribution is a probability distribution where the probability of each symbol is D^{-n} for some positive integer n.

7.5 To reduce $\bar{L}$, we can use blocks of length B. In this case, the source coding theorem gives

$$BH(X) \leq \bar{L}_B < BH(X) + 1 \tag{7.1}$$

$$\implies H(X) \leq \frac{\bar{L}_B}{B} = \bar{L} < H(X) + \frac{1}{B} \tag{7.2}$$

Hence, we can make the expected number of bits per symbol as arbitrarily close to $H(X)$ as possible by varying B.

8 Shannon-Fano-Elias Coding

8.1 **Shannon codes** are those codes that use codewords of length

$$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil \tag{8.1}$$

8.2 The **cumulative distribution function** (CDF) is given by

$$F(x) \triangleq \sum_{z \leq x} p(z) \tag{8.2}$$

8.3 To encode the symbols, we define a modified CDF as follows

$$\bar{F}(x) \triangleq \sum_{z<x} p(z) + \frac{1}{2} p(x) \tag{8.3}$$

Clearly, $x \neq y \iff F(x) \neq F(y)$.

8.4 To represent real $F(x)$ in a finite number of bits, we round it off and use only the first $l(x)$ bits. Thus, if we consider

$$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \tag{8.4}$$

then we can write, using the definition of rounding off,

$$\bar{F}(x) - \left\lfloor \bar{F}(x) \right\rfloor_{l(x)} \leq \frac{1}{2^{l(x)}} \tag{8.5}$$

$$= \frac{1}{2^{\left\lceil \log \frac{1}{p(x)} \right\rceil + 1}} \tag{8.6}$$

$$< \frac{1}{2^{\left(\log \frac{1}{p(x)} + 1\right)}} = \frac{p(x)}{2} \tag{8.7}$$

$$= \bar{F}(x) - F(x-1) \tag{8.8}$$

8.5 Hence, $\left\lfloor \bar{F}(x) \right\rfloor_{l(x)}$ lies between the step corresponding to x, and $l(x)$ bits are sufficient to describe x.

8.6 The interval corresponding to any codeword is of length $2^{-l(x)} < \frac{p(x)}{2}$, which is less than half the step corresponding to x.

8.7 The expected length of the codeword is

$$\bar{L} = \sum_x p(x) l(x) \tag{8.9}$$

$$= \sum_x p(x) \left(\left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \right) \tag{8.10}$$

$$< H(X) + 2 \tag{8.11}$$

This encoding scheme therefore achieves codeword length within two bits of the entropy.

9 Arithmetic Coding

9.1 No restrictions as in Huffman coding to achieve optimality, since we will encode each symbol with an interval in $[0, 1]$.

9.2 Intervals are chosen according to probability, and neseted in the selected interval chosen according to the probability of the symbol shown.

9.3 More likely symbols add less bits to the message since they do not reduce the interval by much.

10 Lempel-Ziv Algorithm

10.1 It is a **universal source coding** algorithm.

10.2 Compress an arbitrary sequence of bits by coding a prefix string from the previously coded strings plus one extra bit. Add this **phrase** as a potential prefix to a dictionary.

10.3 In encoding, specify the location of the prefix phrase in the dictionary and append the new letter. Note that the null prefix has the location 0.

10.4 May not perform well on smaller strings, but for larger documents, the compression approaches optimum.
10.5 Table size must either be large enough, or must contain frequently used prefixes.
10.6 Used in zipping and unzipping files or folders.

11 Run Length Encoding

11.1 Reduce the size of a repeating string of characters, called a **run**.
11.2 Typically each run is encoded in two bytes: the repeating string and the number of repetitions.
11.3 Not high compression ratios, but easy and quick to implement and execute. Used in JPG, TIFF, BMP file formats. Suitable for FAX images.

12 Rate Distortion Function

12.1 Suppose that an analog source $X(t)$ is quantized. Then, let the actual source samples be x_k and the quantized samples be $\tilde{x}_k$.
12.2 The **squared-error distortion** is defined as

$$d(x_k, \tilde{x}_k) \triangleq (x_k - \tilde{x}_k)^2 \tag{12.1}$$

Similar distortion measures could be defined as

$$d(x_k, \tilde{x}_k) \triangleq |x_k - \tilde{x}_k|^p \tag{12.2}$$

12.3 The **Hamming Distortion** is defined as

$$d(x_k, \tilde{x}_k) \triangleq \begin{cases} 0 & x_k = \tilde{x}_k \\ 1 & x_k \neq \tilde{x}_k \end{cases} \tag{12.3}$$

12.4 The **Distortion** between a sequence of n samples X_n and their corresponding quantized values $\tilde{X}_n$ is defined as

$$D \triangleq \mathbb{E}\left[d\left(X_n, \tilde{X}_n\right)\right] \tag{12.4}$$

$$= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[d(x_k, \tilde{x}_k)\right] \tag{12.5}$$

12.5 The **Rate Distortion Function** or the **Information Rate Distortion Function** is the minimum rate (bits/source output) required to represent the output **X** of the memoryless source with a distortion less than or equal to D. It is defined as

$$R(D) \triangleq \min_{p(\tilde{x}|x):\mathbb{E}\left[d\left(X,\tilde{X}\right)\right]\leq D} I\left(X;\tilde{X}\right) \tag{12.6}$$

12.6 Properites of $R(D)$:

a) $R(D)$ is non-increasing in D.

Proof. Suppose $D' \geq D$. Then, by (12.6)

$$R(D) = \min_{p(\tilde{x}|x):\mathbb{E}\left[d\left(X,\tilde{X}\right)\right]\leq D} I\left(X;\tilde{X}\right) \tag{12.7}$$

$$\geq \min_{p(\tilde{x}|x):\mathbb{E}\left[d\left(X,\tilde{X}\right)\right]\leq D'} I\left(X;\tilde{X}\right) \tag{12.8}$$

$$= R(D') \tag{12.9}$$

as required. □

b) $R(D)$ is convex in D.

Proof. Consider two rate distortion pairs (R_i, D_i). Suppose the joint distributions that achieves the minimum rate distortion $R(D_i)$ are p_i, $i \in \{1, 2\}$, where

$$p_i(x, \tilde{x}) = p(x)\, p_i(\tilde{x}|x) \tag{12.10}$$

Define for $\lambda \in [0, 1]$, the joint distribution

$$p_\lambda \triangleq \lambda p_1 + (1-\lambda)\, p_2 \tag{12.11}$$

Since distortion is a linear function of the joint distribution,

$$D_\lambda = \lambda D_1 + (1-\lambda)\, D_2 \tag{12.12}$$

and also since mutual information is convex in the joint distribution,

$$R(D_\lambda) \leq I_{p_\lambda}\left(X;\tilde{X}\right) \tag{12.13}$$

$$\leq \lambda I_{p_1}\left(X;\tilde{X}\right) + (1-\lambda)\, I_{p_2}\left(X;\tilde{X}\right) \tag{12.14}$$

$$= \lambda R(D_1) + (1-\lambda)\, R(D_2) \tag{12.15}$$

as desired. □

c) $R(0) \leq H(X)$

Proof. Setting $\hat{X} = X$, we get $d(X, \hat{X}) = 0$ and so $R(0) \leq H(X)$. □

d) $R(D) = 0$ for $D \geq D_{\max}$.

Proof. Since $I(X;\hat{X}) \geq 0$, we see that $R(D) \geq 0$ for all D. However, $R(D_{\max}) = 0$. Thus, since $R(D)$ is non-increasing, the conclusion follows. □

12.7 The minimum information rate necessary to represent the output of a discrete time continuous amplitudeless memoryless Gaussian source with variance σ_x^2, based on a mean square-error distortion measure per symbol is

$$R_g(D) = \begin{cases} \frac{1}{2}\log\left(\frac{\sigma_x^2}{D}\right) & 0 < D \leq \sigma_x^2 \\ 0 & D > \sigma_x^2 \end{cases} \tag{12.16}$$

12.8 Clearly, $R_g(D)$ is decreasing in D and if $D \geq \sigma_x^2$, then there is no need to transfer any information, since one can use independent zero-mean Gaussian noise samples with variance $D - \sigma_x^2$ for reconstruction.

12.9 There exists and encoding scheme that maps the source output into codewords such that for any given distortion D, the minimum rate $R(D)$ bits per sample is sufficient to reconstruct the source output with an average distortion that is arbitrarily close to D.

12.10 The **Distortion Rate Function** for a discrete time, memoryless Gaussian source is defined as

$$D_g(R) \triangleq 2^{-2R}\sigma_x^2 \tag{12.17}$$

12.11 From (12.17), we can write

$$10\log_{10} D_g(R) = 10\log_{10}\sigma_x^2 - 6R \tag{12.18}$$

thus the mean square distortion decreases at a rate of 6 dB per bit.

13 Optimum Quantizer Design

13.1 We are required to design an optimum scalar quantizer that minimizes some function of the quantization error $q = x - \tilde{x}$, where $\tilde{x}$ is quantized value of x and the amplitude varies according to a probability function $p(x)$. The distortion is given by

$$D = \int_{-\infty}^{\infty} f(\tilde{x} - x)\, p(x)\, dx \tag{13.1}$$

13.2 A *Lloyd-Max quantizer* is one in which the output levels are selected for various input ranges. For an L-levle optimum quantizer, the distortion is given by

$$D = \sum_{k=1}^{L} \int_{x_{k-1}}^{x_k} f(\tilde{x_k} - x) p(x) dx \tag{13.2}$$

13.3 Differentiating (13.2) with respect to $\{x_k\}$ and $\{\tilde{x_k}\}$, we get the system of equations

$$f(\tilde{x_k} - x_k) = f(\tilde{x_{k+1}} - x_k), \ 1 \leq k < L \tag{13.3}$$

$$\int_{x_{k-1}}^{x_k} f'(\tilde{x_k} - x)\, p(x) dx = 0 \tag{13.4}$$

Considering $f(x) = x^2$, we see that

$$x_k = \frac{1}{2}(\tilde{x_k} + \tilde{x_{k+1}}) \tag{13.5}$$

$$\int_{x_{k-1}}^{x_k} (\tilde{x_k} - x)\, p(x) dx = 0 \tag{13.6}$$

13.4 We see that each quantized sample is represented by R bits per sample. However, it is possible to have a more efficient VLC by associating probabilities p_k to each range and using these probabilities to design efficient VLCs.

13.5 To compare performance of different quantizers, we fix distortion D and then compare the average number of bits per sample.

14 Entropy Rate of a Stochastic Process

14.1 A **Stochastic Process** is a rule that associates each outcome of an experiment to a member of a set of random variables, whose distributions may be functions of time.

14.2 A stochastic process is said to be a **Stationary Process** if the joint distribution of any subset of random variables is time-invariant, that is,

$$p(X_1, \ldots, X_n) = p(X_{1+t}, \ldots, X_{n+t}) \tag{14.1}$$

for all $X_i \in X$, $1 \leq i \leq n$ and for all shifts t.

14.3 A discrete stochastic process is said to be a **Markov Chain** or a **Markov Process** if for all $n \geq 1$,

$$p(X_{n+1}|X_1, \ldots, X_n) = p(X_{n+1}|X_n) \tag{14.2}$$

and we denote it as $X_1 \to X_2 \to \ldots \to X_n$. Observe that the reverse order also holds, so we can also denote this as $X_1 \leftrightarrow X_2 \leftrightarrow \ldots \leftrightarrow X_n$.

14.4 The joint distribution of a Markov chain can be written as

$$p(x_1, \ldots, x_n) = p(x_1)\, p(x_2|x_1) \ldots p(x_n|x_{n-1}) \tag{14.3}$$

14.5 The **Data Processing Inequality** states that