Noiseless Source Coding

In this and the next few sections, we shall be concerned with the problems of noiseless data compression. Let $\mathcal{X}$ denote the source alphabet. A binary code $C$ for $\mathcal{X}$ is an encoding $C : \mathcal{X} \rightarrow \{0,1\}^*$ of symbols from $\mathcal{X}$ using the set of finite length binary strings. Binary strings which are images of symbols from $\mathcal{X}$ are called the code-sequences or codewords.

**Definition:** A code $C$ is nonsingular if every element of $\mathcal{X}$ maps to a distinct binary string, i.e., $x_i \neq x_j \implies C(x_i) \neq C(x_j)$.

If a sequence of symbols from $\mathcal{X}$ is given, a code for this sequence may be constructed from a code $C$ by concatenation of the codewords for each symbol in the source sequence.

**Definition:** An extension $C^* : \mathcal{X}^* \rightarrow \{0,1\}^*$ of a code $C$ is a mapping from finite length strings of $\mathcal{X}$ to finite length binary strings defined by

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1) \circ C(x_2) \circ \ldots \circ C(x_n),$$

where “$\circ$” denotes the concatenation operation.

While nonsingularity allows for unique decoding of the codewords when they are presented one at a time, it does not guarantee the unique decodability of the extension, e.g., if $C(a) = 0$ and $C(b) = 00$, then 000 correspond to $aaa$, $ab$ and $ba$.

**Definition:** A code is called uniquely decodable if its extension is nonsingular. A code is called an instantaneous or prefix code if no codeword is a prefix of any other codeword.

Clearly, prefix codes (which should have perhaps been named prefix-free codes) are uniquely decodable – when the string of codewords is scanned from left to right, the first substring which corresponds to a valid codeword can be removed (and decoded), and the
rest of the string processed recursively. We shall, for obvious reasons, be interested only in uniquely decodable codes.

The goal of a source coding scheme is to obtain a compact representation of a sequence of symbols from $\mathcal{X}$. If all the symbols of $\mathcal{X}$ were to occur with about the same frequency, the simple choice of assigning unique indices of approximately $\log |\mathcal{X}|$ bits to each symbol would have been adequate. However, if some symbols are more frequent than others, we may be able to attain a more efficient encoding by assigning shorter codewords to the frequent symbols and (will have to assign) longer codewords to the less frequent ones (to maintain unique decodability). If we use the notation $L(x)$ to denote the length of $C(x)$, the binary string used to represent $x \in \mathcal{X}$, and if $P$ is a probability mass function on $\mathcal{X}$, then the average codeword length is given by $E_P[L(X)] = \sum_{x \in \mathcal{X}} P(x) L(x)$. We shall seek uniquely decodable codes which minimize this expected codeword length.

The next result illustrates why one need not look beyond the class of prefix codes in the search for good uniquely decodable codes. In essence, it says that for any uniquely decodable code, there is a prefix code with identical (or shorter) codeword lengths for every symbol $x \in \mathcal{X}$.

**The Kraft-McMillan Inequality:** The codeword lengths of any uniquely decodable code must satisfy the Kraft inequality

$$\sum_{x \in \mathcal{X}} 2^{-L(x)} \leq 1.$$ 

Conversely, given a set of codeword lengths which satisfy this inequality, it is possible to construct a prefix code with these codeword lengths.

**Proof:** See, e.g., Theorems 5.2.1 and 5.5.1 in T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.

**Redundancy**

If $P$ is a pmf on $\mathcal{X}^n$ and $C_n$ is a binary code for $\mathcal{X}^n$, then the redundancy of $C_n$ with respect to $P$ for a sequence $x_1^n \in \mathcal{X}^n$ is defined as

$$R(x_1^n) = R_{P,C_n}(x_1^n) = L(x_1^n) - \log \frac{1}{P(x_1^n)}.$$ 

Note that since a uniquely decodable satisfies the Kraft inequality,

$$E_P[R(X_1^n)] = \sum_{x_1^n \in \mathcal{X}^n} P(x_1^n) L(x_1^n) - P(x_1^n) \log \frac{1}{P(x_1^n)}$$

2
\[
\begin{align*}
&= E_P[L(X^n)] - H(P) \\
&= \sum_{x^n_1 \in \mathcal{X}^n} P(x^n_1) \log \frac{1}{2^{L(x_1^n)}} + P(x^n_1) \log P(x^n_1) \\
&= \sum_{x^n_1 \in \mathcal{X}^n} P(x^n_1) \log \frac{P(x^n_1)}{2^{L(x_1^n)}} \\
&\geq 1 \log \frac{1}{\sum_{x^n_1 \in \mathcal{X}^n} 2^{-L(x_1^n)}} \geq 0,
\end{align*}
\]

where we have used the log-sum inequality in the last step.

- The entropy \( H(P) \) is a lower bound on the achievable average codeword length for any uniquely decodable code.

- The average redundancy of any uniquely decodable code is nonnegative. It is also clear that to achieve zero redundancy\(^1\), we must be able to choose \( L(x_1^n) = -\log_2 P(x_1^n) \) for every \( x_1^n \in \mathcal{X}^n \).

- The lesser of two problems with achieving zero redundancy is that the logarithms of \( P(x_1^n) \) may not be integer valued for every \( x_1^n \in \mathcal{X}^n \). However, if we choose \( L(x_1^n) = \lfloor -\log_2 P(x_1^n) \rfloor \), then \( E_P[R(X_1^n)] \leq 1 \) bit.

- The more difficult problem with achieving small redundancy is that the pmf \( P \) is often not known to the code designer!

As we shall see shortly, if we encode with \( C_n \) having anything other than these optimal codeword lengths, the total redundancy for coding long sequences using \( C_n \) grows with the length of the sequence, and controlling the rate at which this redundancy grows becomes a crucial factor.

**Remark:** Every (reasonable) uniquely decodable code is a Shannon code with respect to some distribution on the source alphabet. To see this, observe the following.

1. For every uniquely decodable code, there is a (binary) prefix code with the same codeword lengths.

2. The codewords of a binary prefix code may be assigned to nodes in a binary tree in such a manner that the bits of a codeword designate a path from the root to a leaf, and the codeword is identified with this leaf. Every codeword can be assigned to a leaf (not an internal node) and no two codewords get assigned to the same leaf.

\(^1\) A code is said to be a **Shannon code** for a pmf \( P \) on \( \mathcal{X}^n \) if its codeword lengths satisfy \( L(x_1^n) = \lfloor \log \frac{1}{P(x_1^n)} \rfloor \).
3. If the code tree thus produced is not full, i.e., all its leaves are not assigned to codewords, the code can be trivially shortened: for every leaf that is not a codeword, delete the leaf and promote the sibling node to the parent. This does not affect the prefix property of the code, but shortens by one bit all the codewords in the subtree rooted at the sibling thus promoted.

4. The codeword lengths of any uniquely decodable (prefix) code satisfy the Kraft inequality, \( \sum_{x \in \mathcal{X}} 2^{-L(x)} \leq 1 \). Further, if the code tree corresponding to a code is full, then it is easy to show that the inequality is satisfied with equality.

5. The code represented by such a tree is a Shannon code with respect to \( Q(x) = 2^{-L(x)} \).

Henceforth, we shall associate every uniquely decodable (binary) code with such an auxiliary distribution and, e.g., refer to it as the code for the distribution \( Q \). While more than one code may be associated with a \( Q \), this will be of no consequence for our purposes because \( Q \) completely determines the lengths of the codewords.

We have already seen that the average redundancy of any code is nonnegative. However, the redundancy of a code for a particular source symbol (or sequence) may clearly be negative. The next result shows that for random processes the pointwise redundancy \( R(X^n) \) is essentially nonnegative in the sense that it is unlikely to take very large\(^2\) negative values.

**Theorem:** Let \( \{c_n, n = 1, 2, \ldots \} \) be a positive sequence satisfying \( \sum 2^{-c_n} < +\infty \). Then,

\[
R(X^n) \geq -c_n, \quad \text{eventually, almost surely},
\]

i.e., \( \exists \mathcal{B} \subset \mathcal{X}^\infty \), with \( P(\mathcal{B}) = 1 \), and for every \( x_1^\infty \in \mathcal{B} \) there is a \( n_0 = n_0(x_1^\infty) \) such that \( R(x_1^n) \geq -c_n \) for all \( n > n_0 \).

**Proof:** For every \( n \) and \( c > 0 \), let

\[
A_n(c) = \{x_1^n : R(x_1^n) < -c\} = \{x_1^n : 2^{L(x_1^n)}P(x_1^n) < 2^{-c}\},
\]

and note that by construction

\[
P(A_n(c)) = \sum_{x_1^n \in A_n(c)} P(x_1^n) \leq \sum_{x_1^n \in A_n(c)} 2^{-c} 2^{-L(x_1^n)} \leq 2^{-c},
\]

\(^2\)To further develop the notion of what may be large or small in the context of redundancy, note that, in the worst case, we can always encode symbols from \( \mathcal{X}^n \) using binary strings of length \( n \log |\mathcal{X}| \) no matter what the underlying distribution is. Therefore, if the redundancy of a particular code for a particular sequence is \( O(n) \), one may consider the performance of the code for that sequence to be unacceptable. Redundancy of the order of \( n \) is therefore considered “large” for encoding \( \mathcal{X}^n \).
where we have used the Kraft inequality in last step. Therefore

\[ \sum_{n=1}^{\infty} P(R(X_n^n) < -c_n) = \sum_{n=1}^{\infty} P(A_n(c_n)) \leq \sum 2^{-c_n} < +\infty. \]

By the Borel-Cantelli lemma, we can thus conclude that the probability of the events \( A_n(c_n) \) happening infinitely often is zero. In other words, \( R(X_n^n) < -c_n \) only finitely many times and, with probability one, \( R(X_1^n) \geq -c_n \) thereafter. 

**Homework:** For a Shannon code with respect to a measure \( Q \) (on \( X^\infty \)), show that\(^4\) the pointwise redundancy is bounded below by a random variable and \( E[\inf_n R(X_1^n)] \geq - \log_2 e. \)

A notion that must be emphasized here is that \( P \) is a probability measure on the set of infinite sequences with the alphabet \( X \), and \( P(A_n(c)) = P(x_1^\infty : x_1^n \in A_n(c)) \). In general, if \( P \) is a measure on \( X^\infty \), we shall use \( P(x_1^n) \) to designate \( P(x_1^\infty : \hat{x}_1^n = x_1^n) \). Thus the theorem above holds for every sequence of probability measures \( P_n \) which satisfies the Kolmogorov consistency condition: \( \sum_{x_{n+1}} P_{n+1}(x_{n+1}^{n+1}) = P_n(x_1^n) \). Assumptions of ergodicity, stationarity, etc., are not made.

Note that \( c_n = (1 + e) \log n \) satisfies the conditions of the theorem for \( e > 0 \) and, therefore, even the pointwise redundancy of any code with respect to any underlying probability measure is unlikely to be more negative than \( O(\log n) \).

If the sequences to be encoded are sample paths from some known random process \( P \), then we cannot do significantly better in terms of average redundancy than the Shannon code for \( P \). But if \( P \) is not known then, even if we know that \( P \) comes from some parametric family, the redundancy of a Shannon code for a probability measure \( Q \) with respect to the “true” probability \( P \) will go to \( +\infty \) unless \( Q \) is essentially the same as \( P \). This is stated precisely as follows.

**Definition:** Two probability measures \( P \) and \( Q \) are said to be mutually singular if there exists a set \( A \) such that \( P(A) = 1 \) and \( Q(A) = 0 \).

**Theorem:** If \( Q \) is singular with respect to \( P \) then the \( P \)-redundancy of the Shannon code with respect to \( Q \) goes to infinity, \( P \)-almost surely.

---

\(^{3}\)In general, the term “eventually, almost surely” implies that “there is a set of infinite sequences of probability one such that for every sequence in this set, the statement holds for all \( n > n_0 \) where this \( n_0 \) may depend on the sequence, i.e., a uniform \( n_0 \) for the entire set is not required.\(^{3}\)

\(^{4}\)Hint: Bound \( E[\inf_n R(X_1^n)] \) in terms of \( P(\inf_n R(X_1^n) < -c) \), and exploit the relationship between the event \( \inf_n R(x_1^n) < -c \) and the collection \( B_n(c) = \{ x_1^\infty : R(x_1^k) \geq -c, \text{ for every } k < n, \text{ but } R(x_1^n) < -c \}. \)
Lemma: If $P$ and $Q$ are stationary and ergodic probability measures (on $\mathcal{X}^{\infty}$) then either $P = Q$ or $P$ and $Q$ are mutually singular.

We shall omit the proofs of the last two results. The interested reader may contact the instructor for the proofs, which are rather technical in nature. These results are significant because for stationary and ergodic measures distinctness and mutual singularity are synonymous, implying thereby that finite redundancy can be achieved only by knowing the measure exactly!

What is the best that can be done when one does not know the underlying probability measure but has some idea about what it may be?

- What if one knows a reasonable class of measures of which the underlying measure is a member?

- What can be done if the underlying measure is not guaranteed to be in this class, but one can guarantee that the class contains measures which are arbitrarily “close” to it?

These questions shall be answered in precise terms in the next few sections. The loose answer is that for a large category of (underlying) probability measures, the normalized redundancy, $\frac{1}{n} R(X_1^n)$, can almost surely be made to go to zero. Further, there are coding schemes for which the total pointwise redundancy grows as $O(\log n)$, and the normalized redundancy goes to zero, for a class underlying measures. Finally, for any coding scheme which is universal in the sense that for a significant part of a class of underlying measures its normalized redundancy goes to zero, its total redundancy must grow at least at a rate of $\log n$.

Definition: Let $P$ and $Q$ be measures on $\mathcal{X}^{\infty}$, and define $D^{\infty}(P\|Q) = \lim_{n \to \infty} D^n(P\|Q)$, where

$$D^n(P\|Q) = \frac{1}{n} E_P \left[ \log \frac{P(X_1^n)}{Q(X_1^n)} \right],$$

whenever the limit exists. Let $D^{\infty}(P\|Q) = +\infty$ otherwise.

Remark: The limit can be shown to exist whenever $Q$ is a Markov measure of finite memory (for every stationary and ergodic measure $P$). The limit is also well defined if both $P$ and $Q$ are hidden Markov measures with finite state spaces. In the former case, if $Q$ is Markov with memory $k$, the limit can be expressed as function of $Q$ and the $k + 1$ -dimensional marginal of $P$. In the latter, it is expressible only as a limit.
The next results shows that if $P$ can be approximated by finite order Markov measures then the normalized redundancy can be made to go to zero by using a coding scheme which has no prior knowledge of $P$.

**Theorem:** Let $P$ be a stationary and ergodic measure, $\mathcal{U}$ a set of measures, and $\nu$ a probability measure on $\mathcal{U}$ such that for every $\delta > 0$, the set

$$\mathcal{M}(\delta) = \{ U \in \mathcal{U} : U \text{ is a finite order Markov measure with } D^\infty(P\|U) < \delta \}$$

has positive $\nu$-measure. Let $Q$ be a measure obtained by mixing the measures in $\mathcal{U}$ with the “mixture weight” $\nu$ as

$$Q(x^n_1) = \int_{U \in \mathcal{U}} U(x^n_1) \nu(dU).$$

Then the redundancy of the Shannon code with respect to $Q$ satisfies

$$\frac{1}{n} R(X^n_1) \to 0, \quad P\text{-a.s.}$$

**Proof:** Note that since $R(x^n_1) = \log \frac{P(x^n_1)}{Q(x^n_1)}$, the assertion of the theorem is equivalent to the statement that for every $\epsilon > 0$,

$$\log \frac{P(x^n_1)}{Q(x^n_1)} < \epsilon n, \quad \text{eventually, } P\text{-a.s.}$$

i.e., $P(x^n_1) < 2^n Q(x^n_1)$.

Note, however, that for every $\delta > 0$

$$Q(x^n_1) = \int_{U} U(x^n_1) \nu(dU)$$

$$\geq \int_{\mathcal{M}(\delta)} U(x^n_1) \nu(dU), \quad \text{or}$$

$$\frac{2^n}{P(x^n_1)} Q(x^n_1) \geq \int_{\mathcal{M}(\delta)} \frac{2^n U(x^n_1)}{P(x^n_1)} \nu(dU)$$

$$= \int_{\mathcal{M}(\delta)} 2^n \left( -\frac{1}{n} \log \frac{P(x^n_1)}{Q(x^n_1)} \right) \nu(dU).$$

But it can be shown that for a stationary and ergodic measure $P$ and a finite order Markov measure $U \in \mathcal{M}(\delta)$

$$\frac{1}{n} \log \frac{P(X^n_1)}{U(X^n_1)} \to D^\infty(P\|U) < \delta, \quad P\text{-a.s.}$$

and therefore, if we set $\delta = \frac{\epsilon}{3}$, there is a set $\mathcal{B}$ of infinite sequences of $P$-measure 1 such that for every $x_1^\infty \in \mathcal{B}$ the integrand above exceeds $2^n \frac{\epsilon}{3}$ whenever $n > n_0(x_1^\infty, U)$. Since the $\nu$-measure of $\mathcal{M}(\frac{\epsilon}{3})$, the domain of integration, is guaranteed to be positive, the right side of the inequality above goes to $+\infty$ as $n$ increases, for every $x_1^\infty \in \mathcal{B}$. Therefore $P(x^n_1) < 2^n Q(x^n_1)$ eventually, $\forall x_1^\infty \in \mathcal{B}$, and the assertion of the theorem follows. 
