Lecture 6 Source Coding Theorem

October 31 and November 7, 2024

Outline

Question

Let $\pi(n)$ denote the number of primes no greater than n. Note that every positive integer n has a unique prime factorization of the form

$$n = \Pi_{i=1}^{\pi(n)} p_i^{X_i},$$

where $p_1, p_2, \ldots$ are primes, and $X_i = X_i(n)$ is the non-negative integer representing the multiplicity of p_i in the prime factorization of n. Let N be uniformly distributed on $\{1, 2, 3, \ldots, n\}$.

(1) Show that $X_i(N)$ is an integer-valued random variable satisfying

$$0 \leq X_i(N) \leq \log n.$$

(2) Show that

$$\log n = H(N) \leq \pi(n) \log(\log n + 1).$$

Thus not only is $\pi(n) \to \infty$ but in fact $\pi(n) \geq \frac{\log n}{\log(\log n+1)}$.

Proof.

(1) $0 \leq X_i(N)$ is trivial. Note also that $2^{X_i} \leq p_i^{X_i} \leq N \leq n$. Thus, combining both results, $0 \leq X_i(N) \leq \log n$, as we wanted to show.

(2)

$$
\begin{aligned}
\log n &= H(N) \\
&= H(X_1, X_2, \ldots, X_{\pi(n)}) \\
&= \sum_{i=1}^{\pi(n)} H(X_i | X_1, \ldots, X_{i-1}) \\
&\leq H(X_1) + H(X_2) + \ldots + H(X_{\pi(n)}) \\
&= \pi(n) \log(\log n + 1),
\end{aligned}
$$

where the first step follows because there is a one-to-one mapping between N and $X_1, X_2, \ldots, X_{\pi(n)}$. The second step is by the chain rule for entropy. The next step is because conditioning reduced entropy, and the last one is because the distribution that maximizes entropy is the uniform one, there are $\pi(n)$ entropy terms, and X_i's can take at most $\log n + 1$ different values. □

Remark

It is intersting that the same argument applied toa different representation for N *yields a marginally better bound: Suppose we write,*

$$N = M^2 \prod_{p \leq n} p^{Y_p},$$

where $M \geq 1$ *is the largest integer such that* M^2 *divides* N*, and each of the* Y_p *are either zero or one. Then* $H(Y_p) \leq \log 2$ *for all* p*, and the fact that* $M^2 \leq n$ *implies that* $H(M) \leq \log\lfloor\sqrt{n}\rfloor$*. Therefore,*

$$\begin{aligned}
\log n = H(N) &= H(M, Y_{p_1}, Y_{p_2}, \cdot, Y_{p_{\pi(n)}}) \\
&\leq H(M) + \sum_{p \leq n} H(Y_p) \\
&\leq \frac{1}{2}\log n + \pi(n)\log 2,
\end{aligned}$$

which implies that $\pi(n) \geq \frac{\log n}{2\log 2}$*, for all* $n \geq 2$*.*

Outline

A file is composed of a sequence of types. A byte is composed of 8 bits and can have a decimal value between 0 and 255. A typical text file is composed of the ASCII character set (decimal values 0 to 127). This character set uses only seven of the eight bits in a byte.

A file is composed of a sequence of types. A byte is composed of 8 bits and can have a decimal value between 0 and 255. A typical text file is composed of the ASCII character set (decimal values 0 to 127). This character set uses only seven of the eight bits in a byte.

Question

By how much could the size of a file be reduce given that it is an ASCII file? How would you achieve this reduction?

One way of measuring the information content of a random variable is simply to count the number of possible outcomes, $|\mathcal{A}_X|$. If we gave a binary name to each outcome, the length of each name would be $\log_2 |\mathcal{A}_X|$ bits, if $|\mathcal{A}_X|$ happened to be a power of 2.

One way of measuring the information content of a random variable is simply to count the number of possible outcomes, $|\mathcal{A}_X|$. If we gave a binary name to each outcome, the length of each name would be $\log_2 |\mathcal{A}_X|$ bits, if $|\mathcal{A}_X|$ happened to be a power of 2.

Definition

The raw bit content of X is

$$H_0(X) = \log_2 |\mathcal{A}_X|.$$

Question

Could there be a compressor that maps an outcome x to a binary code $c(x)$, and a decompressor that maps c back to x, such that every possible outcome is compressed into a binary code of length shorter than $H_0(X)$ bits?

You can not give $\mathcal{A}_X$ outcomes unique binary names of some length l shorter than $\log_2 |\mathcal{A}_X|$ outcomes unqiuely binary names of some length l shorter than $\log_2 |\mathcal{A}_X|$ bits, because there are only 2^l such binary names, and $l < \log_2 |\mathcal{A}_X|$ implies $2^l < |\mathcal{A}_X|$, so at least two different inputs to the compressor would compress to the sme output file.

Outline

- Whichever type of compressor we construct, we need somehow to take into account the probabilities of the different outcomes.
- Imagine comparing the information contents of two text files – one in which all 128 ASCII characters are used with equal probability, and one in which the characters are used with their frequencies in English text.
- Can we define a measure of information content that distinguishes between these two files?
- Intuitively, the latter file contains less information per character because it is more predictable.

- One simple way to use our knowledge that some symbols have a smaller probability is to imagine recoding the observations into a smaller alphabet – thus losing the ability to encode some of the more improbable symbols – and then measuring the raw bit content of the new alphabet.
- For example, we might take a risk when compressing English text, guessing that the most infrequent characters won't occur, and make a reduced ASCII code that omits the characters $\{!, @, \#, \%, \wedge, *, \sim, <, >, /, \backslash,, \{, \}, [,], |\}$, thereby reducing the size of the alphabet by seventeen.
- The larger the risk we are willing to take, the smaller our final alphabet becomes.

Example

Let

$$\mathcal{X} = \{a, b, c, d, e, f, g, h\}$$

and

$$P_X = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{3}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}.$$

The raw bit content of this ensemble is 3 *bits, corresponding to* 8 *binary names. But notice that* $P(x \in \{a, b, c, d\}) = 15/16$. *So if we are willing to run a risk of* $\delta = 1/16$ *of not having a name for* x*, then we can get by four names - half as many names as are needed if every* $x \in \mathcal{X}$ *has a name.*

$\delta = 0$	
x	$c(x)$
a	000
b	001
c	010
d	011
e	100
f	101
g	110
h	111

$\delta = 1/16$	
x	$c(x)$
a	00
b	01
c	10
d	11
e	—
f	—
g	—
h	—

Definition

The smallest δ-sufficient subset S_δ is the smallest subset of $\mathcal{A}_X$ satisfying

$$P(x \in S_\delta) \geq 1 - \delta.$$

The subset S_δ can be constructed by ranking the elements of $\mathcal{A}_X$ in order of decreasing probability and adding successive elements until the total probability is $\geq (1 - \delta)$.

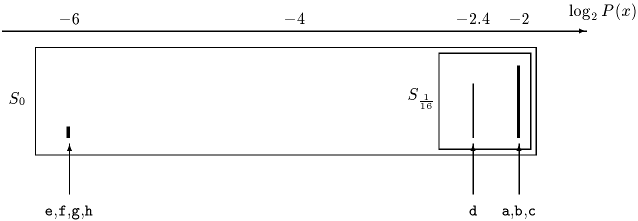
- Let us now formalize this idea.
- To make a compression strategy with risk δ, we make the smallest possible subset S_δ such that the probability that x is not in S_δ is less than or equal to δ, i.e., $P(x \notin S_\delta) \leq \delta$.
- For each value of δ we can then define a new measure of information content - the $\log$ of the size of this smallest subset S_δ.
- In ensembles in which several elements have the same probability, there may be several smallest subsets that contain different elements, but all that matters is their sizes (which are equal), so we will not dwell on this ambiguity.

Definition

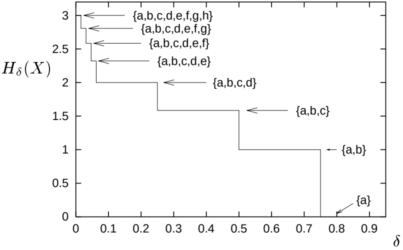
The essential bit content of X is

$$H_\delta(X) = \log_2 |S_\delta|.$$

Note that $H_0(X)$ is the special case of $H_\delta(X)$ with $\delta = 0$ (if $P(x) > 0$ for all $x \in \mathcal{A}_X$.


−6
−4
−2.4
−2
$\log_2 P(x)$
S_0
$S_{\frac{1}{16}}$
e,f,g,h
d
a,b,c

(a)


$H_\delta(X)$
{a,b,c,d,e,f,g,h}
{a,b,c,d,e,f,g}
{a,b,c,d,e,f}
{a,b,c,d,e}
{a,b,c,d}
{a,b,c}
{a,b}
{a}
0
0.5
1
1.5
2
2.5
3
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
δ

(b)

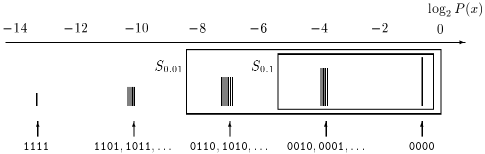
We now turn to examples where the outcome $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is a string of N independent identically distributed random variables from a single random variable X. We will denote by X^N the random vector $(X_1, X_2, \ldots, X_n)$. Remember that entropy is additive for independent variables, so $H(X^N) = NH(X)$.

Example

Consider a string of N flips of a bent coin, $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, where $x_n \in \{0, 1\}$, with probabilities $p_0 = 0.9$, $p_1 = 0.1$. If $r(\boldsymbol{x})$ is the number of 1s in $\boldsymbol{x}$ then

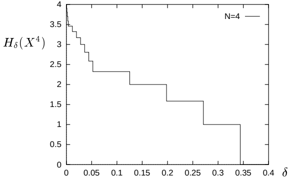
$$P(\boldsymbol{x}) = p_0^{N - r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}.$$

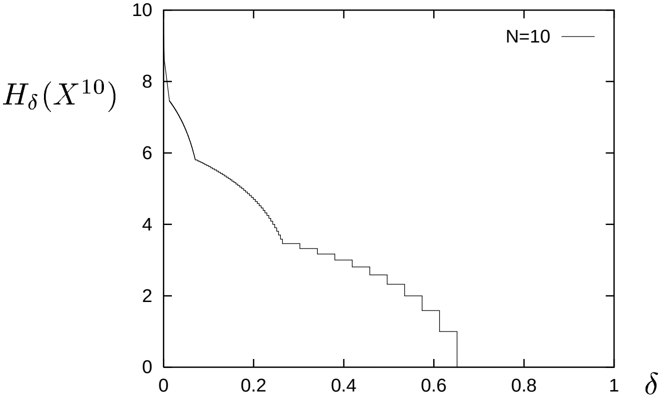
To evaluate $H_\delta(X^N)$ we must find the smallest sufficient subset S_δ. This subset will contain all $\mathbf{X}$ with $r(\mathbf{x}) = 0, 1, 2, \cdots$, up to some $r_{\max}(\delta) - 1$, and some of the $\mathbf{x}$ with $r(\mathbf{x}) = r_{\max}(\delta)$.

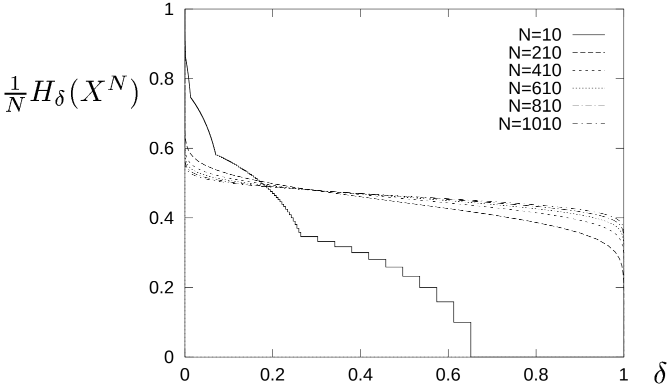

$\log_2 P(x)$
−14
−12
−10
−8
−6
−4
−2
0
$S_{0.01}$
$S_{0.1}$
1111
1101, 1011, ...
0110, 1010, ...
0010, 0001, ...
0000

(a)

(b)


$H_\delta(X^4)$
N=4
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
δ


$H_\delta(X^{10})$
N=10
10
8
6
4
2
0
0
0.2
0.4
0.6
0.8
1
δ


$\frac{1}{N}H_\delta(X^N)$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
δ
N=10
N=210
N=410
N=610
N=810
N=1010

Theorem (Shannon's source coding theorem)

Let X be an random variable with entropy $H(X) = H$ bits. Given $\varepsilon > 0$ and $0 < \delta < 1$, there exists a positive integer N_0 such that for $N > N_0$,

$$|\frac{1}{N} H_\delta(X^N) - H| < \varepsilon.$$

Outline

Why does increasing N help? Let's examine long strings from X^N.

Why does increasing N help? Let's examine long strings from X^N.

Let us consider the case of coin flip problem introduced in last lecture, where $N = 100$ and $p_1 = 0.1$.

Why does increasing N help? Let's examine long strings from X^N.

Let us consider the case of coin flip problem introduced in last lecture, where $N = 100$ and $p_1 = 0.1$.

The probability of a string **x** that contains r 1s and $N - r$ 0s is

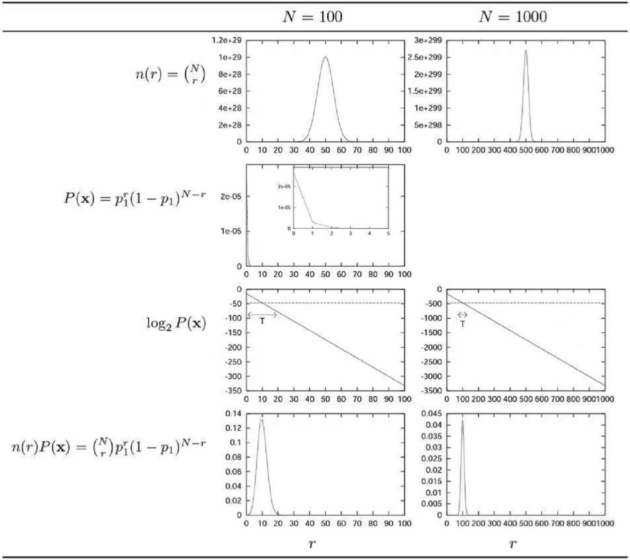
$$P(\mathbf{x}) = p_1^r (1 - p_1)^{N-r}.$$

The number of strings that contain r 1s is

$$n(r) = \binom{N}{r}.$$

So the number of 1s, r, has a binomial distribution:

$$P(r) = \binom{N}{r} p_1^r (1 - p_1)^{N-r}.$$


N = 100
N = 1000
$n(r) = \binom{N}{r}$
$P(\mathbf{x}) = p_1^r(1-p_1)^{N-r}$
$\log_2 P(\mathbf{x})$
$n(r)P(\mathbf{x}) = \binom{N}{r} p_1^r(1-p_1)^{N-r}$
T
T
r
r

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

A long string of N symbols will usually contain about $p_1 N$ occurrences of the first symbol, $p_2 N$ occurrences of the second, etc.

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

A long string of N symbols will usually contain about p_1N occurrences of the first symbol, p_2N occurrences of the second, etc.

The probability of this string is roughly

$$p(\mathbf{x})_{typ} = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \approx p_1^{p_1N} p_2^{p_2N} \cdots p_I^{p_IN}$$

so that the information cotent of atypical string is

$$\log_2 \frac{1}{P(\mathbf{x})} \approx N \sum_i p_i \log_2 \frac{1}{p_i} = NH.$$

Definition

We call the set typical elements *the typical set*, $T_{N,\beta}$:

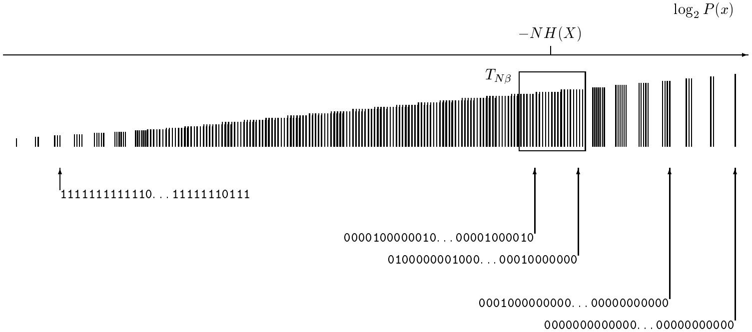
$$T_{N,\beta} := \{\mathbf{x} \in \mathcal{X}^N : |\frac{1}{N}\log_2\frac{1}{P(\mathbf{x})} - H| < \varepsilon\}.$$

Asymptotic equipartition property

For an ensemble of N independent identically distributed random variables $X^N := (X_1, X_2, \cdots, X_N)$, with N sufficiently large, the outcome $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is almost certain to belong to a subset of $\mathcal{X}^N$ having only $2^{NH(X)}$ members, each having probability 'close to' $2^{-NH(X)}$.

The difference between the smallest δ-sufficient subset and the typical set

Consider coin flip problem again. The typical sequences in this case are the sequence in which the proportion of 0's is close to 0.9. However, this does not includes the sequence of all 0's, which is the most likely single sequence. The smallest δ-sufficient subset includes all the most probable sequences and therefore includes the sequence of all 0's.


$\log_2 P(x)$
$-NH(X)$
$T_{N\beta}$
1111111111110. . . 11111110111
0000100000010. . . 00001000010
0100000001000. . . 00010000000
0001000000000. . . 00000000000
0000000000000. . . 00000000000

Why do we introduce the typical set?

The best choice of subset for block compression is (by definition) S_δ, not a typical set. So why did we bother introducing the typical set?

Why do we introduce the typical set?

The best choice of subset for block compression is (by definition) S_δ, not a typical set. So why did we bother introducing the typical set?

The answer is, we can count the typical set.

Outline

Theorem (Weak law of large numbers)

Let $X_1, \cdots, X_n$ be N independent random variables, having common mean μ and common variance σ^2. Then

$$P((\frac{1}{N}\sum_{i=1}^{N} X_i - \mu)^2 \geq \alpha) \leq \sigma^2/\alpha N.$$

We again define the typical set with parameters N and β (In the textbook, it is denoted by $A_\epsilon^{(n)}$) thus:

$$T_{N,\beta} := \{\mathbf{x} \in \mathcal{X}^N : |\frac{1}{N}\log_2\frac{1}{P(\mathbf{x})} - H| < \beta\}.$$

For all $\mathbf{x} \in T_{N,\beta}$, the probability of $\mathbf{x}$ satisfies

$$2^{-N(H+\beta)} < p(\mathbf{x}) < 2^{-N(H-\beta)}.$$

So from the weak law of large numbers, we have that

$$P(\mathbf{x} \in T_{N,\beta}) \geq 1 - \frac{\sigma^2}{\beta^2 N}.$$

- We have thus proved the 'asymptotic equipartition' principle. As N increases, the probability that $\mathbf{x}$ falls in $T_{N,\beta}$ approaches 1, for any β.
- How does this result relate to source coding?
- We must relate $T_{N,\beta}$ to $H_\delta(X^N)$.
- We will show that for any given δ there is a sufficiently big N such that $H_\delta(X^N) \simeq NH$.

Step 1. $\frac{1}{N} H_\delta(X^N) < H + \varepsilon$ when N is large enough.

- The set $T_{N,\beta}$ is not the best subset for compression. So the size of $T_{N,\beta}$ gives an upper bound on H_δ. We shall show how small $h_\delta(X^N)$ must be by calculating how big $T_{N,\beta}$ could possibly be.
- The smallest possible probability that a member of $T_{N,\beta}$ can have is $2^{-N(H+\beta)}$, and the total probability contained by $T_{N,\beta}$ can't be any bigger than 1. So $|T_{N,\beta}| 2^{-N(H+\beta)} < 1$, that is, the size of the typical set is bounded by

$$|T_{N,\beta}| < 2^{N(H+\beta)}.$$

- If we set $\beta = \varepsilon$ and N_0 such that $\frac{\sigma^2}{\varepsilon^2 N_0} \leq \delta$, then $P(\mathbf{x} \in T_{N,\beta}) \geq 1 - \delta$, and the set $T_{N,\beta}$ becomes a witness to the fact that $H_\delta(X^N) \leq \log_2 |T_{N,\beta}| < N(H + \varepsilon)$.

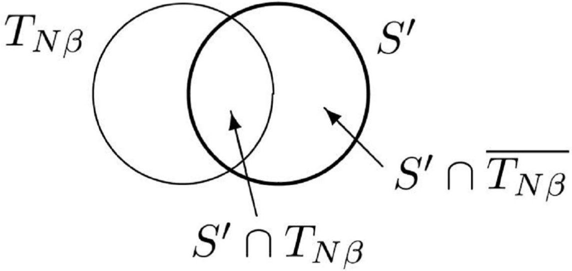
Step 2. $\frac{1}{N} H_\delta(X^N) > H - \varepsilon$ when N is large enough.

Imagine that someone claims this is not so, which means that for any N, the smallest δ-sufficient S_δ is smaller then the above inequality would allow. We can make use of our typical set to show that they must be mistaken.

Step 2. $\frac{1}{N} H_\delta(X^N) > H - \varepsilon$ when N is large enough.

Imagine that someone claims this is not so, which means that for any N, the smallest δ-sufficient S_δ is smaller then the above inequality would allow. We can make use of our typical set to show that they must be mistaken.

Remember that we are free to set β to any value we choose. We will set $\beta = \varepsilon/2$, so that our task is to prove that a subset S' having $|S'| \leq 2^{N(H-2\beta)}$ and achieving $P(\mathbf{x} \in S') \geq 1 - \delta$ cannot exist (for N greater than an N_0 that we will specify).



So, let us consider the probability of falling in this rival smaller subset S'. The probability of the subset S' is

$$P(\mathbf{x} \in S') = P(\mathbf{x} \in S' \cap T_{N,\beta}) + P(\mathbf{x} \in S' \cap \overline{T_{N,\beta}}),$$

where $\overline{T_{N,\beta}}$ denotes the complement $\{\mathbf{x} \notin T_{N,\beta}\}$.

- Now we have that

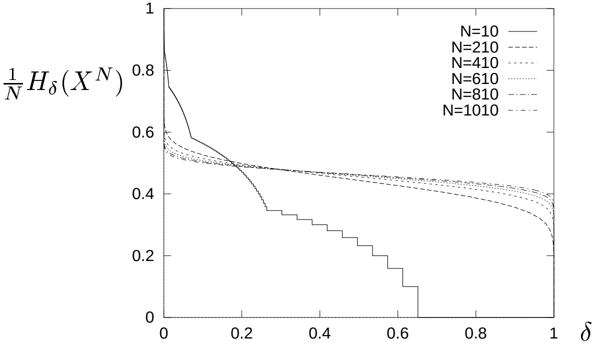
$$P(\mathbf{x} \in S') = P(\mathbf{x} \in S' \cap T_{N,\beta}) + P(\mathbf{x} \in S' \cap \overline{T_{N,\beta}}).$$

- The maximum value of the first term is found if $S' \cap T_{N,\beta}$ contains $2^{N(H-2\beta)}$ outcomes all with the maximum probability, $2^{-N(H-\beta)}$.
- The maximum value of the first term can have is $P(\mathbf{x} \notin T_{N,\beta})$. So:

$$P(\mathbf{x} \in S') \leq 2^{N(H-2\beta)} 2^{-N(H-\beta)} + \frac{\sigma^2}{\beta^2 N} = 2^{-N\beta} + \frac{\sigma^2}{\beta^2 N}.$$

- We can now set $\beta = \varepsilon/2$ and N_0 such that $P(\mathbf{x} \in S') < 1 - \delta$, which shows that S' cannot satisfy the definition of a sufficient subset S_δ.
- Thus any subset S' with size $|S'| \leq 2^{N(H-\varepsilon)}$ has probability less than $1 - \delta$, so by the definition of H_δ, $H_\delta > N(H - \varepsilon)$.

Thus for large enough N, the function $\frac{1}{N}H_\delta(X^N)$ is essentially a constant function of δ.



Remarks

The source coding theorem has two parts, $\frac{1}{N}H_\delta(X^N) < H + \varepsilon$, and $\frac{1}{N}H_\delta(X^N) > H - \varepsilon$. Both results are interesting.

- The first part tells us that even if the probability of error δ is extremely small, the number of bits per symbol $\frac{1}{N}H_\delta(X^N)$ needed to specify a long N-symbol string **x** with vanishingly small error probability does not have to exceed $H + \varepsilon$ bits. We need to have only a tiny tolerance for error, and the number of bits required drops significantly from $H_0(X)$ to $H + \varepsilon$.
- What happens if we are yet more tolerant to compression errors? The proof of the second part tells us that if we are using the typical set to code, even δ is very close to 1, so that errors are made most of the time, the average number of bits per symbol needed to specify **x** must still be at least $H - \varepsilon$ bits.
- These two extreme tells us that regardless of our specify **x** is H bits; no more or no less.

- In we use variable-length compression, we can archive the same compression rate while it is not lossy. Check Theorem 3.2.1 in the textbook.
- Let X_1, X_2, $\cdots$, X_N be independent, identically distributed random variables drawn from the probability mass function $p(x)$.
- We order all the elements in each set according to some order.
- Then we can represent each sequence of the typical set $T_{\beta,N}$ by giving the index of the sequence in the set.

- Since there are $\leq 2^{N(H+\beta)}$ sequences in $T_{\beta,N}$, the indexing requires no more than $N(H+\beta)+1$ bits.
- We prefix all these sequences by a 0, giving a total length of $\leq N(H+\beta)+2$ bits to represent each sequence $T_{N,\beta}$.
- Similarly, we can index each sequence not in $T_{N,\beta}$ by using not more than $n \log |\mathcal{X}| + 1$ bits.
- Prefixing these indices by 1, we have a code for all sequences in $\mathcal{X}^n$.

- We use the notation x^N to denote the sequence $x_1, x_2, \cdots, x_N$.
- Let $l(x^N)$ be the length of the codeword corresponding to x^N.
- If N is sufficiently large so that $P(T_{N,\beta}) \geq 1-\beta$, the expected length of the codeword is

$$
\begin{aligned}
&\mathsf{E}(l(X^N)) \\
=&\sum_{x^N} p(x^N) l(x^N) = \sum_{X^N \in T_{N,\beta}} p(x^N) l(x^N) + \sum_{X^N \notin T_{N,\beta}} p(x^N) l(x^N) \\
\leq& \sum_{X^N \in T_{N,\beta}} p(x^N)(N(H+\beta)+2) + \sum_{X^N \notin T_{N,\beta}} p(x^N)(N \log |\mathcal{X}| + 2) \\
\leq& N(H+\beta) + \beta N(\log |\mathcal{X}|) + 2 \\
=& N(H+\varepsilon),
\end{aligned}
$$

where $\varepsilon = \beta + \beta \log |\mathcal{X}| + \frac{2}{N}$ can be made arbitrarily small by an appropriate choice of N.

Theorem

Let X^n be i.i.d. $\sim p(x)$. Let $\varepsilon > 0$. Then there exists a code that maps sequences x^n of length n into binary strings such that the mapping is one-to-one (and therefore invertible) and

$$E[\frac{1}{n}l(X^n)] \leq H(X) + \varepsilon$$

for n sufficiently large.

The compression scheme described in the proof is impractical. From the next lecture, we shall discuss practical compression algorithms.

Remarks

- The AEP for ergodic processes has come to be known as the **Shannon-McMillan-Breiman theorem**. In this lecture we have proven the AEP for i.i.d. processes.
- In fact, AEP holds for general ergodic processes.
- An ergodic source is defined on a probability space $(\Omega, \mathcal{B}, P)$, where $\mathcal{B}$ is a σ-algebra of subsets of Ω and P is a probability measure.
- We also have a transformation $T : \Omega \to \Omega$, which plays the role of a time shift.

- We will say that the transformation is **stationary** if $P(TA) = P(A)$ for all $A \in \mathcal{B}$.
- The transformation is called **ergodic** if every set A such that $TA = A$ a.e., satisfies $P(A) = 0$ or 1.
- IF T is stationary and ergodic, we say that the process defined by $X_n(\omega) = X(T^n\omega)$ is **stationary and ergodic**.
- For a stationary ergodic source, Birkhoff's ergodic theorem states that

$$\frac{1}{n}\sum_{i=1}^{n} X_i(\omega) \to \mathsf{E}X = \int X dP \text{ with probability } 1.$$

- Thus, the law of large numbers holds for ergodic processes.

Shannon-McMillan-Breiman theorem

Theorem

If H is the entropy rate of a finite-valued stationary ergodic process $\{X_n\}$, then

$$-\frac{1}{n}\log p(X_0, \cdots, X_{n-1}) \rightarrow H \text{ with probability } 1.$$