### Extensions of a Source

We formalize the notion of encoding symbols in blocks by defining the $N$-th extension of a source, in which we look at sequences of symbols, written as $(X_1, \ldots, X_N)$ or $X^N$.

If our original source alphabet, $\mathcal{A}_X$, has $I$ symbols, the source alphabet for its $N$-th extension, $\mathcal{A}_X^N$, will have $I^N$ symbols — all possible blocks of $N$ symbols from $\mathcal{A}_X$.

If the probabilities for symbols in $\mathcal{A}_X$ are $p_1, \ldots, p_I$, the probabilities for symbols in $\mathcal{A}_X^N$ are found by multiplying the $p_i$ for all the symbols in the block. (This is appropriate when symbols are independent.)

For instance, if $N = 3$:

$$P(X_1, X_2, X_3) = (a_i, a_j, a_k) = p_{ij} p_{jk}$$

### Entropy of an Extension

We now prove that $H(X^N) = NH(X)$:

$$H(X^N) = \sum_{i_1=1}^I \cdots \sum_{i_N=1}^I p_{i_1} \cdots p_{i_N} \log \left( \frac{1}{p_{i_1} \cdots p_{i_N}} \right)$$

$$= \sum_{i_1=1}^I \cdots \sum_{i_N=1}^I p_{i_1} \cdots p_{i_N} \sum_{j=1}^N \log \left( \frac{1}{p_{i_j}} \right)$$

$$= \sum_{j=1}^N \sum_{i_1=1}^I \cdots \sum_{i_N=1}^I p_{i_1} \cdots p_{i_N} \log \left( \frac{1}{p_{i_j}} \right)$$

$$= \sum_{j=1}^N \sum_{i_j=1}^I p_{i_j} \log \left( \frac{1}{p_{i_j}} \right)$$

$$\times \sum_{i_k \neq j} p_{i_1} \cdots p_{i_{j-1}} p_{i_{j+1}} \cdots p_{i_N}$$

$$= \sum_{j=1}^N \sum_{i_j=1}^I p_{i_j} \log \left( \frac{1}{p_{i_j}} \right) = NH(X)$$

(Or just use the fact that $E(U + V) = E(U) + E(V)$.)

### Shannon's Noiseless Coding Theorem

By using extensions of the source, we can compress arbitrarily close to the entropy!

Formally:

For any desired average length per symbol, $R$, that is greater than the binary entropy, $H(X)$, there is a value of $N$ for which a uniquely decodable binary code for $X^N$ exists that has expected length less than $NR$.

### Proof of Shannon's Noiseless Coding Theorem

Consider coding the $N$-th extension of a source whose symbols have probabilities $p_1, \ldots, p_I$, using an binary Shannon-Fano code.

The Shannon-Fano code for blocks of $N$ symbols will have expected codeword length, $L_N$, no greater than $1 + H(X^N) = 1 + NH(X)$.

The expected codeword length per original source symbol will therefore be no greater than

$$\frac{L_N}{N} = \frac{1 + NH(X)}{N} = H(X) + \frac{1}{N}$$

By choosing $N$ to be large enough, we can make this as close to the entropy, $H(X)$, as we wish.
Another Way to Compress Down to the Entropy

We get a similar result by supposing that we will always encode \( N \) symbols into a block of exactly \( NR \) bits. Can we do this in a way that is very likely to be decodable?

Yes, for large values of \( N \). As discussed in Section 4.3 of MacKay’s book, the Law of Large Numbers tells us that the sequence of symbols to encode, \( a_1, \ldots, a_N \), is very likely to be a “typical” one, for which

\[
\frac{1}{N} \log_2(1/(p_1 \cdots p_N)) = \frac{1}{N} \sum_{j=1}^{N} \log_2(1/p_i)
\]

is very close to the expectation of \( \log_2(1/p_i) \), which is the entropy, \( H(X) = \sum_i p_i \log_2(1/p_i) \).

So if we encode all the sequences in this typical set in a way that can be decoded, the code will almost always be uniquely decodable.

How Big is the Typical Set?

Let’s define “typical” sequences as ones where

\[
(1/N) \log_2(1/(p_1 \cdots p_N)) \leq H(X) + \eta/\sqrt{N}
\]

We scale the margin allowed above \( H(X) \) as \( 1/\sqrt{N} \) since that’s how the standard deviation of an average scales. Chebychev’s inequality then tells us that most sequences will satisfy this inequality, if \( \eta \) is set to a fairly large value.

The probability of any such typical sequence will satisfy

\[
p_1 \cdots p_N \geq 2^{-NH(X) - \eta/\sqrt{N}}
\]

The total probability for all such sequences can’t be greater than one, so the number of “typical” sequences can’t be greater than

\[
2^{NH(X) + \eta/\sqrt{N}}
\]

We will be able to encode these sequences in \( NR \) bits if \( NR \geq N H(X) + \eta \sqrt{N} \). If \( R > H(X) \), this will be true if \( N \) is sufficiently large.

An End and a Beginning

Shannon’s Noiseless Coding Theorem is mathematically satisfying. From a practical point of view, though, we still have two problems:

- How can we compress data to nearly the entropy in practice?
  The number of possible blocks of size \( N \) is \( I^N \) — huge when \( N \) is large. And \( N \) sometimes must be large to get close to the entropy by encoding blocks of size \( N \).
  One solution: A technique known as arithmetic coding.

- Where do the symbol probabilities \( p_1, \ldots, p_I \) come from? And are symbols really independent, with known, constant probabilities?
  This is the problem of source modeling.