Side Note: Algorithmic Information Content of Symbol Sequences and Marginal Entropies

36-462, Spring 2009

9 February 2009, in connection with lecture 8

Consider a long sequence of discrete symbols $x_1, x_2, \ldots x_n \equiv x_1^n$. Each symbol can take one of k distinct values, say to be concrete the whole numbers $0, 1 \ldots k-1$. How can we bound $K(x_1^n)$, the algorithmic information content of the string?

Let's start with the $k = 2$ case, where the book-keeping is simplest. If we think back to elementary probability (or elementary combinatiorics), we realize that there are only so many ways of combining a given number of zeroes and a given number of ones. Suppose we knew that there were n_0 zeroes and $n_1 = n - n_0$ ones in the sequence. The number of such sequences is $\binom{n}{n_0}$, a.k.a. a "binomial coefficient". We could order the sequences however we like, and then encode the given string by saying where it falls in that ordering, i.e., that enumeration. The number of bits needed for this is the log of binomial coefficient. Of course we also have to say what n_0 is, but the allowed values are just the integers from 0 to n, so that takes $\log_2 n + 1$ bits. So

$$K(x_1^n) \leq \log_2 \binom{n}{n_0} + \log_2 n + 1 + c$$

where c is the overhead for recovering the sequence from its serial number, printing, etc. So what can we say about the log of the binomial coefficients?

Well,

$$\binom{n}{n_0} = \frac{n!}{n_0! n_1!}$$

and there is Stirling's approximation for $\log q!$:

$$\log q! = q \log q + o(q)$$

so

$$\begin{aligned}
\log \binom{n}{n_0} &= \log n! - \log n_0! - \log n_1! + o(n) \\
&= n \log n - n_0 \log n_0 - n_1 \log n_1 + o(n) \\
&= n_0 \log n + n_1 \log n - n_0 \log n_0 - n_1 \log n_1 + o(n)
\end{aligned}$$

$$
\begin{aligned}
&= -n_0 \log n_0/n - n_1 \log n_1/n + o(n) \\
&= -n\frac{n_0}{n} \log \frac{n_0}{n} - n\frac{n_1}{n} \log \frac{n_1}{n} + o(n) \\
&= -np_0 \log p_0 - np_1 \log p_1 + o(n) \\
&= nH(p_0) + o(n)
\end{aligned}
$$

where $p_i = n_i/n$ is the actual relative frequency of the symbol i in x_1^n, and $H(p)$ is the Shannon entropy of a binary variable where one symbol has probability p. Let's plug this back in to our bound on the algorithmic information:

$$K(x_1^n) \leq nH(p_0) + o(n)$$

where the remainder term $o(n)$ has absorbed $\log_2 n + 1 + c$. Since $H(p) = 1$ when and only when $p = 1/2$, if the relative frequencies of the two symbols are not equal, then the sequence is compressible.

What if $k > 2$? The same kind of reasoning applies. If we know $n_0, n_1, \ldots n_{k-1}$, then there are

$$\frac{n!}{\prod_{i=0}^{k-1} n_i!}$$

possible sequences. Specifying the counts takes at most $k \log n + 1$ bits[1]. So

$$K(x_1^n) \leq c + k \log n + 1 + \log n! - \sum_{i=1}^{k-1} \log n_i!$$

Pulling the same trick with Stirling's approximation, we get

$$K(x_1^n) \leq nH(p_0, p_1, \ldots p_{k-1}) + o(n)$$

where now $H()$ is the entropy of a random k-valued multinomial random variable with a given distribution. In this case the distribution is just $p_i = n_i/n$.

EXERCISE: step through the algebra.

The same argument works for symbol pairs. Start with the binary case once again. Divide x_1^n into $n/2$ non-overlapping blocks of two symbols. (If n is not even, we just need at most one extra bit to encode the last symbol, which will be negligible.) Each of these must be either 00, 01, 10, or 11. This is however just the previous case, with an alphabet of size $k = 4$, so that

$$K(x_1^n) \leq \frac{n}{2} H(p_{00}, p_{01}, p_{10}, p_{11}) + o(n)$$

in the obvious notation.[2] But the same trick will work with an arbitrary-sized alphabet k and and arbitrary block sizes m:

$$K(x_1^n) \leq \frac{n}{m} H[\hat{X}_1^m] + o(n)$$

[1] If we were really worried about this, we could use the fact that $\sum n_i = n$ to reduce this encoding length, but we're about to absorb everything which isn't at least linear in n into a remainder term.

[2] To see this, note that the number of ways of arranging these blocks is

$$\frac{(n/2)!}{n_{00}!n_{01}!n_{10}!n_{11}!}$$

where $\hat{X}_1^m$ is a (fictitious) random variable which follows the distribution of length-m blocks from x_1^n.

Two things seem to be worth noting here.

1. If we had a stationary sequence of random variables $X_1, \ldots X_n$, it would be true that
$$H(X_1^n) \leq \frac{n}{m} H[X_1^m]$$
with equality if and only if successive length-m blocks were statistically independent. For fixed m, as n grows, $H[\hat{X}_1^m] \rightarrow H[X_1^m]$ by ergodicity, so the algorithmic information of long sequences should be upper bounded by the entropy of blocks from that sequence. It turns out that it can't be much lower, either, *on average*.

2. The naive or literal encoding length of x_1^n is $n \log k$ bits. The sequence is incompressible if its algorithmic information content is close to its literal length. But notice that $H[\hat{X}_1^m] \leq m \log k$. In fact, recall from the slides on defining entropy in terms of relative entropy that $H[\hat{X}_1^m] = m \log k - D(U \| \hat{P}^{(m)})$ where U is the uniform distribution and $\hat{P}^{(m)}$ is the distribution of m-blocks in x_1^n. So the sequence will be compressible unless all *blocks* are uniformly distributed.

Of course if we let m get large enough, for fixed n, the remainder terms we've buried in $o(n)$ come back to haunt us.

---

with the obvious notation for the counts of the blocks, and of course the adding-up constraint that $n_{00} + n_{01} + n_{10} + n_{11} = n/2$. Now we get

$$\log \frac{(n/2)!}{n_{00}! n_{01}! n_{10}! n_{11}!} = -\frac{n}{2} \left[p_{00} \log p_{00} + p_{01} \log p_{01} + p_{10} \log p_{10} + p_{11} \log p_{11} \right]$$

which is what we'd get from a four-symbol alphabet.