Markov Sources

An *K-th order Markov source* is one in which the the probability of a symbol depends on the preceding K symbols.

We can write the probability of a sequence of symbols, $X_1, X_2, \ldots, X_n$ from such a source as follows (for $K = 2$):

$$
\begin{aligned}
& P(X_1 = s_{i_1}, X_2 = s_{i_2}, \ldots, X_n = s_{i_n}) \\
&= P(X_1 = s_{i_1}) \times P(X_2 = s_{i_2} \mid X_1 = s_{i_1}) \\
&\quad \times P(X_3 = s_{i_3} \mid X_1 = s_{i_1}, X_2 = s_{i_2}) \\
&\quad \times P(X_4 = s_{i_4} \mid X_2 = s_{i_2}, X_3 = s_{i_3}) \\
&\quad \cdots \\
&\quad \times P(X_n = s_{i_n} \mid X_{n-2} = s_{i_{n-2}}, X_{n-1} = s_{i_{n-1}}) \\
&= P(X_1 = s_{i_1}) \times P(X_2 = s_{i_2} \mid X_1 = s_{i_1}) \\
&\quad \times M(i_1, i_2, i_3) M(i_2, i_3, i_4) \cdots M(i_{n-2}, i_{n-1}, i_n)
\end{aligned}
$$

Here, $M(i, j, k)$ is the probability of symbol s_k when the preceding two symbols were s_i and s_j.

Adaptive Markov Models

Some sources may really be Markov of some order K, but usually not.

We can nevertheless use a Markov *model* for a source as the basis for data compression.

Usually, we don't know what the "transition probabilities", should be, so we estimate them adaptively. Eg, for $K = 2$, we accumulate frequencies in each context, $F(i, j, k)$, and then use probabilities

$$M(i, j, k) = F(i, j, k) / \sum_{k'} F(i, j, k')$$

After encoding symbol s_k in context s_i, s_j, we increment $F(i, j, k)$.

A K-th order Markov model has to handle the first $K-1$ symbols specially. One approach: Imagine that there are K symbols before the beginning with some special value (eg, space).

Markov Models of Order 0, 1, and 2 Applied to English Text

I applied adaptive Markov models of order 0, 1, and 2, using arithmetic coding, to three English text files (Latex), of varying sizes.

Markov Model of Order 0

Uncompressed file size	Compressed file size	Compression factor	Bits per character
2344	1431	1.64	4.88
20192	12055	1.67	4.78
235215	137284	1.71	4.67

Markov Model of Order 1

Uncompressed file size	Compressed file size	Compression factor	Bits per character
2344	1750	1.34	5.97
20192	11490	1.76	4.55
235215	114494	2.05	3.89

Markov Model of Order 2

Uncompressed file size	Compressed file size	Compression factor	Bits per character
2344	2061	1.14	7.03
20192	13379	1.51	5.30
235215	111408	2.11	3.79

How Large an Order Should be Used?

We can see a problem with these results.

A Markov model of high order works well with long files, in which most of the characters are encoded after good statistics have been gathered.

But for small files, high-order models don't work well — most characters occur in contexts that have occurred only a few times before, or never before.

For the smallest file, the zero-order model with only one context was best, even though we know that English has strong dependencies between characters!