Optimal source code

Now we proved that any codeword set that satisfies the prefix condition has to satisfy the Kraft inequality,

and that the Kraft inequality is a sufficient condition for the existence of a codeword set with the specified set of codeword lengths.

We now consider the problem of finding the prefix code with the minimum expected length.

This is equivalent to finding the set of lengths l_1, l_2, . . . , l_m satisfying the Kraft inequality and whose expected length $L = \sum p_i l_i$ is less than the expected length of any other prefix code.

This is a standard optimization problem: Minimize L over all integers l_1, l_2, . . . , l_m satisfying Kraft inequality over the l_i's.

Bounds on Optimal Code Length

We now demonstrate a code that achieves an expected length L within 1 bit of the lower bound; that is, $H(X) \leq L < H(X) + 1$

We wish to minimize $L = \sum p_i l_i$ subject to the constraint that l_1, l_2, . . . , l_m are integers and $\sum r^{-l_i} \leq 1$.

We proved that the optimal codeword lengths can be found by finding the probability distribution closest to the distribution of X in relative entropy,

p_i may not equal an integer, we round it up to give integer word-length assignments,

$l_i = \lceil log_r \frac{1}{p_i} \rceil$

These lengths satisfy the Kraft inequality since

$\sum r^{-\lceil log\frac{1}{p_i}\rceil} \leq \sum r^{-log\frac{1}{p_i}} = \sum p_i = 1$

This choice of codeword lengths satisfies

Let $Q_i = \frac{r^{-l_i}}{\sum_{i=1}^{q} r^{-l_i}}$

$log_r\frac{1}{p_i} \leq l_i < log_r\frac{1}{p_i} + 1$

Multiplying by p_i and summing over i, we obtain $H_r(X) \leq L < H_r(X) + 1$

Since an optimal code can only be better than this code, we have the following theorem.

Theorem: Let $l_1, l_2, \ldots, l_m$ be optimal codeword lengths for a source distribution p and a r-ary alphabet, and let L be the associated expected length of an optimal code ($L = \sum p_i l_i$).

Then $H_r(X) \leq L < H_r(X) + 1$

Shannon First Theorem

$H_r(S) \leq L < H_r(S) + 1$

Then $H_r(S^n) \leq L_n < H_r(S^n) + 1$ or, $H_r(S) \leq \frac{L_n}{n} < H_r(S) + \frac{1}{n}$

In the limit, $\lim_{n \to \infty} \frac{L_n}{n} = H_r(S)$

This is the noiseless coding theorem which suggests that by coding the n-th extension of S, one can make the average number of r-ary code symbols per source symbol as small as possible but not smaller than the entropy of the source. $\frac{L_n}{n}$ better approximates the average codeword length.

For Markov source, the adjoint will obey the bound on L. i.e. $H_r(\bar{S}) \leq L$

Augmenting previous results,

$H_r(S) \leq H_r(\bar{S}) \leq L$

and $H_r(S^n) \leq H_r(\bar{S^n}) \leq L_n$

Now select l_i as unique integer satisfying

$log_r \frac{1}{P_i} \leq l_i < log_r \frac{1}{P_i} + 1$ - so that

$H_r(S) + \frac{H_r(\bar{S}) - H_r(S)}{n} \leq \frac{L_n}{n} < H_r(S) + \frac{H_r(\bar{S}) - H_r(S) + 1}{n}$

Here also $\lim_{n \to \infty} \frac{L_n}{n}$ can be made as close to $H_r(S)$ as possible.

Shannon Fano coding scheme

The length assignment described above gives rise to Shannon-Fano code. So, lenghts are chosen as $log_r(\frac{1}{P_i}) \leq l_i < log_r(\frac{1}{P_i}) + 1$

The problem with this scheme is that it does not consider the relative positions of symbols with respect to one another.

It only considers absolute probabilities. Ex. $p_A = 1/2^{10}$ and $p_B = 1 - 1/2^{10}$ gives $l_A = 10$ and $l_B = 1$.

From common sense, it appears that the choice should be l_A and l_B should both be 1.

However, in the long run, the Shannon-Fano assignment would not increase the average length too much. Nevertheless, we should do better.

Huffman coding scheme

Efficiency of code is given by $\eta = \frac{H_r(S)}{L}$

Redundancy of code is given by $1-\eta = \frac{H_r(S)-L}{L}$

An optimal (shortest expected length) prefix code for a given distribution can be constructed by a simple algorithm discovered by Huffman.

First, sort the probabilities $P_1 \geq P_2 \geq \ldots \geq P_q$

Reduce the source to q-1 symbols and reorder (sort again) until no of symbols =2.

Reduced sequence S_j has s_α which has s_{α_0} and s_{α_1} in S_{j-1}.

Then $P_\alpha = P_{\alpha_0} + P_{\alpha_1}$

At every level, if the two symbols having smallest probabilities are collapsed into a compound symbol and we go on constructing the coding tree as a heap, we get the optimal code.

Then, overall average length would increase by an amount $L_{j-1} = L_j + P_{\alpha_0} + P_{\alpha_1}$; since only these two symbols have increased length.

So, the overhead due to expansion of a compound symbol is minimum if the smallest probability symbols are chosen for the purpose.

Any other choice of s_{α_0} and s_{α_1} would not be optimal.

This exhibits a greedy choice and therefore at every stage of heap formation, we should sort the symbol probabilities and collapse the two symbols with smallest probabilities as we go up towards the root of the coding tree.

Sensitivity of Huffman coding scheme

Suppose the probability assignment used for code compression is different from what occurs in real life.

$p'_i = p_i + e_i$ such that

$\frac{1}{q}\sum_{i=1}^{q} e_i = 0$ and $var(e_i) = \sigma^2 = \frac{1}{q}\sum e_i^2$

Hence, $L' = \frac{1}{q}\sum l_i p'_i = L + \frac{1}{q}\sum l_i e_i$

We should examine $\frac{1}{q}\sum l_i e_i$ to get a better feel of how the length is affected by noise.

Using Lagrange multipliers λ and μ,

$\mathcal{L} = \frac{1}{q} l_i e_i - \lambda \frac{1}{q}\sum e_i - \mu(\frac{1}{q}\sum e_i^2 - \sigma^2)$

$\frac{\partial \mathcal{L}}{\partial e_i} = \frac{1}{q}(l_i - \lambda - 2\mu e_i) = 0$

On summing over i, $\frac{1}{q}\sum_i l_i - \lambda = 0$

gives $\lambda = \frac{1}{q}\sum_i l_i$.

$e_i \frac{\partial \mathcal{L}}{\partial e_i} = 0$ gives

$\frac{1}{q}\sum l_i e_i - \frac{2\mu}{q}\sum(e_i^2) = 0$ to get $\mu = \frac{\sum e_i l_i}{2q\sigma^2}$.

Now, putting λ and μ in $l_i \frac{\partial \mathcal{L}}{\partial e_i} = 0$ gives

$\frac{1}{q}\sum l_i^2 - \lambda \frac{1}{q}\sum l_i - 2\mu \frac{1}{q}\sum(e_i l_i) = 0$

Then, we can write $(\frac{1}{q}\sum e_i l_i)^2 = (\frac{1}{q}\sum l_i^2 - (\frac{1}{q}\sum l_i)^2)\sigma^2$

i.e. $(\frac{1}{q}\sum e_i l_i)^2 =$ Var (l_i) Var (e_i)

This implies that high variance of codeword lengths make the average length of code more prone to variation with noise.

Comparing some coding schemes

Symbol Space	Prob	Code-I	Code-II	Code-III
A	.5	0	00	0111
B	.3	10	01	011
C	.1	110	10	01
D	.1	111	11	0

Expected length (Code-I) = 1.7 (uniquely decodable and instantaneous)

For Code-II it is 2.0 (fixed length, easy to decode)

For Code-III it is 3.2 (uniquely decodable, not instantaneous; not efficient as well).