Kraft's and McMillan's Inequalities

For the purpose of compression, the only interesting property of a code besides being uniquely decodable is the lengths of the codewords. Kraft's inequality gives an exact condition for the existence of a prefix code in terms of the codeword lengths.

Theorem 1.11: (Kraft's Inequality) There exists a binary prefix code with codeword lengths $\ell_1, \ell_2, \ldots, \ell_\sigma$ if and only if

$$\sum_{i=1}^{\sigma} 2^{-\ell_i} \leq 1 \ .$$

Proof. Consider a binary search on the real interval $[0,1)$. In each step, the current interval is split into two halves and one of the halves is chosen as the new interval. We can associate a search of ℓ steps with a binary string of length ℓ: zero corresponds to choosing the left half and one to choosing the right half. For any binary string B, let $I(B)$ be the final interval of the associated search. For example, 1011 corresponds to the search sequence $[0,1), [1/2,2/2), [2/4,3/4), [5/8,6/8), [11/16,12/16)$ and $I(B) = [11/16, 12/16)$.

Now consider the set $\{I(w) \mid w \in \{0,1\}^\ell\}$ of all intervals corresponding to binary strings of lengths ℓ. Clearly, an interval $[l,r)$ belongs to this set if and only if

- The length $r - l$ of the interval is $2^{-\ell}$.
- Both end points l and r are multiples of $2^{-\ell}$.

Such intervals are called *dyadic*. For any two binary strings w and w', the intervals $I(w)$ and $I(w')$ overlap if and only if one of the strings is a prefix of the other.

Now we are ready to prove the theorem starting with the "only if" part. Let C be a prefix code with the codewords $w_1, w_2, \ldots, w_\sigma$ of lengths $\ell_1, \ell_2, \ldots, \ell_\sigma$. Since C is a prefix code, the intervals $I(w_i)$, $i \in [1..\sigma]$, do not overlap with each other, and their total length is at most 1. Thus

$$\sum_{i=1}^{\sigma} |I(w_i)| = \sum_{i=1}^{\sigma} 2^{-\ell_i} \leq 1 \ .$$

Finally, we prove the "if" part. Let $\ell_1, \ell_2, \ldots, \ell_\sigma$ be arbitrary integers satisfying the inequality. We will construct a prefix code with those codeword lengths. We start by sorting the lengths in nondecreasing order, i.e., we assume that $\ell_1 \leq \ell_2 \leq \ldots \ell_\sigma$. For $i \in [1..\sigma]$, let

$$[l_i, r_i) = \left[\sum_{j=1}^{i-1} 2^{-\ell_j}, \sum_{j=1}^{i} 2^{-\ell_j}\right)$$

For example, the lengths $1,2,3$ would result the intervals $[0,1/2), [1/2,3/4), [3/4,7/8)$. The intervals $[l_i, r_i)$ are dyadic:

- The length of the interval $[l_i, r_i)$ is $2^{-\ell_i}$.
- Both end points l_i and r_i are multiples $2^{-\ell_i}$. This is because, for all $j < i$, $\ell_j \leq \ell_i$ and thus $2^{-\ell_j}$ is a multiple of $2^{-\ell_i}$.

Thus $[l_i, r_i) = I(w_i)$ for some binary string w_i of length ℓ_i. Since the intervals do not overlap, the binary strings are not prefixes of each other. Thus $w_1, w_2, \ldots, w_\sigma$ are valid codewords for a prefix code with the required lengths. $\square$

Prefix codes are a subclass of uniquely decodable codes. McMillan's inequality generalizes the result for all uniquely decodable codes.

Theorem 1.12: (McMillan's Inequality) There exists a uniquely decodable binary code with codeword lengths $\ell_1, \ell_2, \ldots, \ell_\sigma$ if and only if

$$\sum_{i=1}^{\sigma} 2^{-\ell_i} \leq 1 \ .$$

Proof. Omitted.

The Huffman algorithm and most other methods for constructing codes are restricted to prefix codes. Together the two inequalities show that this is not a significant restriction with respect to compression.

Corollary 1.13: For any uniquely decodable binary code, there exists a binary prefix code with the same codeword lengths.

In particular, the Huffman code is optimal among all uniquely decodable codes.

Using Kraft's inequality, we can also characterize redundancy in prefix codes.

Definition 1.14: A prefix code satisfying Kraft's inequality with strict inequality ($\sum 2^{-\ell_i} < 1$) is called *redundant*. A prefix code satisfying Kraft's inequality with strict equality ($\sum 2^{-\ell_i} = 1$) is called *complete*.

Corollary 1.15: For any redundant prefix code with codeword lengths $\ell_1, \ell_2, \ldots, \ell_\sigma$, there exists a complete prefix code with codeword lengths $\ell'_1, \ell'_2, \ldots, \ell'_\sigma$ such that $\ell'_i \leq \ell_i$ for all $i \in [1..\sigma]$.

Proof. Assume ℓ_σ is the longest codeword length in the redundant code. For all $i \in [1..\sigma]$, $2^{-\ell_i}$ is a multiple of $2^{-\ell_\sigma}$. Thus the redundancy gap $1 - \sum 2^{-\ell_i}$ is a multiple of $2^{-\ell_\sigma}$ too. If we reduce ℓ_σ by one, we increase the sum $\sum 2^{-\ell_i}$ by $2^{-\ell_\sigma}$ and still satisfy Kraft's inequality. If the resulting code is still redundant, repeat the procedure until it becomes complete.
□

Integer codes

Next we look at some variable-length codes for integers. We want to have a prefix code when the source alphabet is the set $\mathbb{N}$ of natural numbers.

We cannot use the Huffman algorithm or anything similar on an infinite alphabet. Often there is an upper bound on the integers, but even then the alphabet can be too large to compute and store the code efficiently.

Instead, we have a few fixed codes for all natural numbers. With each of them, the codeword length is a non-decreasing function of the integer value. The codes differ in how fast the growth is.

The codes described here are for all non-negative integers including zero. If the minimum integer value n_{min} is larger than zero, we can shift the codes, i.e., use the code for $n - n_{min}$ to encode n. The case $n_{min} = 1$ is common in the literature.

Let us first define the standard binary codes that are used as a building block in the other codes. This is not a prefix code for natural numbers but a family of fixed-length codes for finite integer ranges.

Definition 1.16: (Binary Code) For any $k \geq 0$ and $n \in [0..2^k - 1]$

$$\text{binary}_k(n) = k\text{-bit binary representation of } n.$$

Our first proper prefix code is the unary code.

Definition 1.17: (Unary code) For any $n \in \mathbb{N}$

$$\text{unary}(n) = 1^n 0$$

The length of the unary code for n is $n + 1$.

Example 1.18: $\text{binary}_4(6) = 0110$ and $\text{unary}(6) = 1111110$.

The unary code lengths grow too fast for many purposes. Elias γ and δ codes achieve a logarithmic growth rates.

Definition 1.19: (Elias γ and δ code) For any $n \in \mathbb{N}$

$$\gamma(n) = \text{unary}(k) \cdot \text{binary}_k(n - 2^k + 1)$$
$$\delta(n) = \gamma(k) \cdot \text{binary}_k(n - 2^k + 1)$$

where $k = \lfloor \log(n + 1) \rfloor$.

The lengths of the codes are

$$|\gamma(n)| = 2\lfloor \log(n + 1) \rfloor + 1$$
$$|\delta(n)| = \lceil \log(n + 2) \rceil + 2\lfloor \log \lceil \log(n + 2) \rceil \rfloor$$

Example 1.20: If $n = 6$, then $k = 2$ and $\gamma(6) = 110\ 11$ and $\delta(6) = 101\ 11$.

Golomb–Rice codes are a family of prefix codes GR_0, GR_1, GR_2, ... with a codeword length growth rate between unary code and Elias codes.

Definition 1.21: (Golomb–Rice codes) For any $k \geq 0$ and any $n \in \mathbb{N}$

$$\mathsf{GR}_k(n) = \mathsf{unary}(q) \cdot \mathsf{binary}_k(n - q2^k)$$

where $q = \lfloor n/2^k \rfloor$.

Note that GR_k for a fixed

The codeword lengths are

$$|\mathsf{GR}_k(n)| = \lfloor n/2^k \rfloor + k + 1$$

Example 1.22: $\mathsf{GR}_1(6) = 1110\,0$, $\mathsf{GR}_2(6) = 10\,10$, $\mathsf{GR}_3(6) = 0\,110$

There are many possible variants and extension of the codes we have seen.

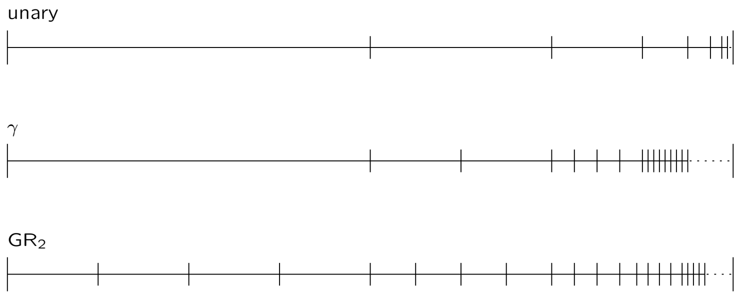
Here are some examples of the codewords.

n	unary(n)	$\gamma(n)$	$\delta(n)$	$\mathsf{GR}_2(n)$	$\mathsf{GR}_4(n)$
0	0	0	0	000	00000
1	10	100	1000	001	00001
2	110	101	1001	010	00010
3	1110	11000	10100	011	00011
4	11110	11001	10101	1000	00100
5	111110	11010	10110	1001	00101
6	1111110	11011	10111	1010	00110
7	11111110	1110000	11000000	1011	00111
8	111111110	1110001	11000001	11000	01000
9	1111111110	1110010	11000010	11001	01001
10	111...10	1110011	11000011	11010	01010
20	111...10	111100101	110010101	11111000	100100
30	111...10	111101111	110011111	1111111010	101110
40	111...10	11111001001	1101001001	1111111111000	1101000
50	111...10	11111010011	1101010011	111...1010	11100010
60	111...10	11111011101	1101011101	111...1000	11101100

All of these integer codes are complete prefix codes. For example,

$$\sum_{n \in \mathbb{N}} 2^{-|\gamma(n)|} = 1.$$

Example 1.23: The intervals corresponding to the codewords (as in the proof of Kraft's inequality) of length at most 7 for unary, γ and GR_2 codes.



The unused part at the end is the "code space" available for all the longer codes.