Formal Modeling in Cognitive Science

Lecture 28: Kraft Inequality; Source Coding Theorem; Huffman Coding

Frank Keller

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

March 13, 2006

Kraft Inequality

Problem: construct an instantaneous code of minimum expected length for a given random variable. The following inequality holds:

Theorem: Kraft Inequality

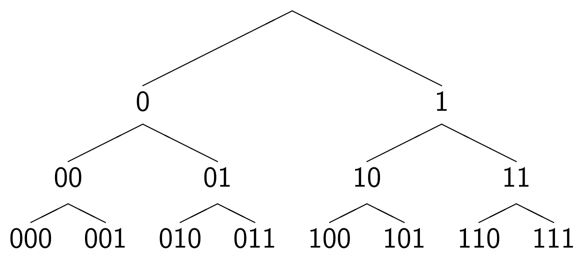
For an instantaneous code C for a random variable X, the code word lengths $l(x)$ must satisfy the inequality:

$$\sum_{x \in X} 2^{-l(x)} \leq 1$$

Conversely, if the code word lengths satisfy this inequality, then there exists an instantaneous code with these word lengths.

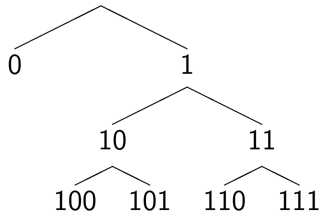
Kraft Inequality

We can illustrate the Kraft Inequality using a *coding tree.* Start with a tree that contains all three-bit codes:



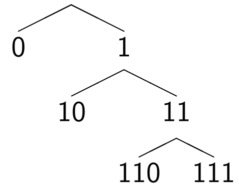
Kraft Inequality

For each code word, prune all the branches below it (as they violate the prefix condition). For example, if we decide to use the code word 0, we get the following tree:



Kraft Inequality

Now if we decide to use the code word 10:



The remaining leaves constitute a prefix code. Kraft inequality:

$$\sum_{x \in X} 2^{-l(x)} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} = 1$$

Shannon Information

The Kraft inequality tells us that an instantaneous code exists. But we are interested in finding the *optimal* code, i.e., one that minimized the *expected code length $L(C)$*.

Theorem: Shannon Information

The expected length $L(C)$ of a code C for the random variable X with distribution $f(x)$ is minimal if the code word lengths $l(x)$ are given by:

$$l(x) = -\log f(x)$$

This quantity is called the Shannon information.

Shannon information is *pointwise entropy.* (See mutual information and pointwise mutual information.)

Shannon Information

Example

Consider the following random variable with the optimal code lengths given by the Shannon information:

x	a	b	c	d
$f(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
$l(x)$	1	2	3	3

The expected code length $L(C)$ for the optimal code is:

$$L(C) = \sum_{x \in X} f(x) l(x) = -\sum_{x \in X} f(x) \log f(x) = 1.75$$

Note that this is the same as the entropy of X, $H(X)$.

Lower Bound on Expected Length

This observation about the relation between the entropy and the expected length of the optimal code can be generalized:

Theorem: Lower Bound on Expected Length

Let C be an instantaneous code for the random variable X. Then the expected code length $L(C)$ is bounded by:

$$L(C) \geq H(X)$$

Upper Bound on Expected Length

Of course we are more interested in finding an upper bound, i.e., a code that has a maximum expected length:

Theorem: Source Coding Theorem

Let C a code with optimal code lengths, i.e, $l(x) = -\log f(x)$ for the random variable X with distribution $f(x)$. Then the expected length $L(C)$ is bounded by:

$$H(X) \leq L(C) < H(X) + 1$$

Why is the upper bound $H(X) + 1$ and not $H(X)$? Because sometimes the Shannon information gives us fractional lengths; we have to round up.

Source Coding Theorem

Example

Consider the following random variable with the optimal code lengths given by the Shannon information:

x	a	b	c	d	e
$f(x)$	0.25	0.25	0.2	0.15	0.15
$l(x)$	2.0	2.0	2.3	2.7	2.7

The entropy of this random variable is $H(X) = 2.2855$. The source coding theorem tells us:

$$2.2855 \leq L(C) < 3.2855$$

where $L(C)$ is the code length of the optimal code.

Source Coding Theorem

Example

Now consider the following code that tries to the code words on the optimal code lengths as closely as possible:

x	a	b	c	d	e
$C(x)$	00	10	11	010	011
$l(x)$	2	2	2	3	3

The expected code length for this code is therefore $L(C) = 2.30$. This is very close to the optimal code length of $H(X) = 2.2855$.

Huffman Coding

The source coding theorem tells us the properties of the optimal code, but not how to find it. A number of algorithms exists for this.

Here, we consider *Huffman coding*, an algorithm that constructs a code with the following properties:

- instantaneous (prefix code);
- optimal (shortest expected length code).

The expected code length of the Huffman code is bounded by $H(X) + 1$.

Huffman Coding

1. Find the two symbols with the smallest probability and combine them into a new symbol and add their probabilities.
2. Repeat step (2) until there is only one symbol left with a probability of 1.
3. Draw all the symbols in the form of a tree which branches every time two symbols are combined.
4. Label all the left branches of the tree with a 0 and all the right branches with a 1.
5. The code for a symbol is the sequence of 0s and 1s that lead to it on the tree, starting from the root (with probability 1).

Huffman Coding

Example

Assume we want to encode the set of all vowels, and we have the following probability distribution:

x	a	e	i	o	u
$f(x)$	0.12	0.42	0.09	0.30	0.07
$-\log f(x)$	3.06	1.25	3.47	1.74	3.84

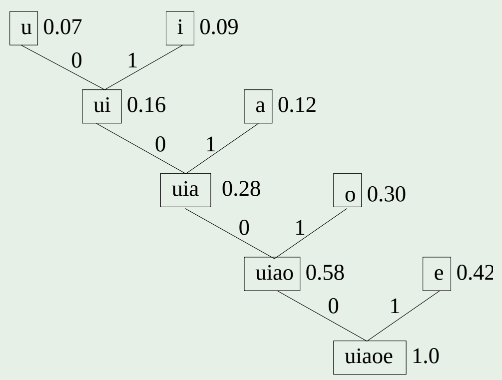
The Huffman code for this distribution is:

x	a	e	i	o	u
$C(x)$	001	1	0001	01	0000
$l(x)$	3	1	4	2	4

Generate this code by drawing the Huffman coding tree.

Huffman Coding

Example



Summary

- The optimal length of a code word is given by its Shannon information: $-\log f(x)$;
- source coding theorem: the expected length of the optimal code is bounded by entropy: $H(X) \leq L(C) < H(X) + 1$.
- Huffman Coding is an algorithm for finding an optimal instantaneous code for a given random variable.