Foundations of Computer Security

Lecture 33: Entropy II

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Example Revisited

Given: an unbalanced coin that is three times more likely to yield a head than a tail.

Solution: There are two possible outcomes:

Result	**Prob**
H	3/4
T	1/4

The entropy is computed as follows:

$$h = -(3/4 \times \log 3/4 + 1/4 \times \log 1/4) \approx 0.811$$

Upshot: It's theoretically impossible to encode this language using less than 0.811 bits per symbol (on average).

But how would you ever do better than 1 bit / symbol?

Example Revisited

Instead of taking single flips as our "experiments," let's take pairs (call them "2flips") and code as follows:

Result	Prob.	Code
HH	$9/16$	0
HT	$3/16$	10
TH	$3/16$	110
TT	$1/16$	111

Suppose we flip the coin 32 times; that's 16 2flips. In an *average* run, we'd expect: HH to appear 9 times, HT and TH to each appear 3 times, and TT to appear once. *Why?*

Example Revisited

Given 32 flips (16 2flips), we could expect:

Result	Count	Code	Bits
HH	9	0	9
HT	3	10	6
TH	3	110	9
TT	1	111	3
Total:			27

For the naïve encoding, using 1 bit / flip, we'd expect to use 32 bits. Our efficiency is $27/32 \approx 0.844$, which is not a bad approximation of the entropy (0.811).

Could we do better? Sure, just use 3flips, 4flips, etc. The entropy is the limit of this process.

Test Your Understanding

Suppose you have a six-sided die that is unbalanced such that 1 and 2 are equally likely; 3 and 4 are equally likely; and 5 and 6 are equally likely. However, the die rolls 1 twice as often as 3, and rolls 3 three times as often as 5.

1. What is the "naive" encoding for this language?
2. What is the entropy of this language?
3. Find an encoding that is more efficient than the naive encoding.
4. Give a convincing argument that your encoding is more efficient than the naive encoding.

Hint: There's no need to encode sequences of rolls.

Lessons

- Computing the entropy of a language provides a bound on the efficiency of any encoding.
- But, finding an efficient encoding requires ingenuity.

Next lecture: Fundamental Theorems