Information theory: HW #2 solution

Table of Contents

- 1. Markov chain entropy
- 2. Huffman codes

1 Markov chain entropy

First, short remark I've already given in the previous homework: I'm putting some haskell code here just for fun/more detailed explanation/proof-of-work. Please skip it if it's too borkng.

Let's consider the following matrix:

$$E = \begin{pmatrix} 1/4 & 0 & 3/4 \\ 0 & 1/4 & 3/4 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}$$

First task is to find such p that $p = pH$. This can be done, in fact, in two ways. First one is solving the following equation:

$$\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} x & y & z \end{pmatrix} * E$$

Indeed, this equation system has a solution (skipped): $P = (1/6, 1/3, 1/2)$. But it can be though obtained in another way. Let's suppose E is ergodic, then accordingly to ergodic theorem [1] E^n converges to A, where each row of it is equal to P. This can be easily seen in practice:

```
-- | This function takes matrix and raises it to the power n.
-- (!*!) is matrix multiplication.
(!*^) :: Matrix -> Int -> Matrix
(!*^) m 0 = m
(!*^) m i = m !*! (m !*^ (i-1))

-- Type aliases for vector and matrix
type Vector = V.Vector Double
type Matrix = V.Vector Vector

-- | Calculate the entropy
entropy :: Vector -> Double
entropy = sum . V.map (\p -> - p * log2 p)

-- | Given matrix
matrix :: Matrix
matrix =
    V.fromList $ map V.fromList $
    [[1/4,  0, 3/4],
    [  0, 1/4, 3/4],
    [1/4, 1/2, 1/4]]

-- | Solution vector calculated from raising matrix E to the power
-- 5000, takes 0'th row.
p :: Vector
p = (matrix !*^ 5000) ! 0
```

If printed, value of `P` is `[0.16666666666666663,0.3333333333333333,0.5000000000000001]`, that seems pretty much to convert to the exact solution.

The entropy of the random source with given probabilities: $H = 1.459147$ (calculated as `entropy p`).

Now let's calculate $H(X|X^\infty)$:

$$H(X|X^\infty) = H(X|X) = -\sum_i P_i \sum_j P_{ij} log(P_i j)$$

$P_i j$ is exactly E. Proof of the first equality can be found in the course textbook (1.7, example 1.7.2). The value calculated is $H(X|X^\infty) = 1.155639$. Now we are able to compute $H_n(X)$ using the formula in the end of chapter 1.7 of the textbook:

$$H_n(X) = H(X|X^n) + \frac{s}{n}(H_s(X) - H(X|X^s))$$

We'll use a fact that $H(X|X^n)$ equals to $H(X|X)$ and already computed. Plus the markov's chain is simple, so $s = 1$. Thus formula looks like this:

$$H_n(X) = H(X|X) + \frac{1}{n}(H(X) - H(X|X)) = 1.1556 + \frac{0.303}{n}$$

And so $H_2 = 1.1556 + 0.303/2 = 1.3071$.

2 Huffman codes

Given three words a, b, c with probabilities $\frac{1}{6}, \frac{1}{3}, \frac{1}{2}$, we can assign the following codes to them:

Word	Probability	Code
a	0.166	01
b	0.333	00

Word	Probability	Code
c	0.5	1

Then it's easy to see that average amount of bits required to encode one word is $2 * 0.166 + 2 * 0.333 + 1 * 0.5 = 1.498$, which is more than $H(X)$. Let's then consider blocks of size 2 and calculate $P(XY) = P(X) * P(X|Y)$, where second probability is from E:

Word	P(X)	P(X|Y)	P(XY)	Code
bc	0.333	0.75	0.250	00
cb	0.5	0.5	0.250	01
ac	0.166	0.75	0.125	101
ca	0.5	0.25	0.125	110
cc	0.5	0.25	0.125	111
bb	0.333	0.25	0.083	1001
aa	0.166	0.25	0.042	10001
ab	0.166	0	0	100000
ba	0.333	0	0	100001

And then average bit amount per word is:

$$\frac{0.250 * 2 * 2 + 0.125 * 3 * 3 + 0.083 * 4 + 0.042 * 5}{2} = \frac{2.667}{2} = 1.334$$

, which appears to be more than 1.498 that we got in the previous attempt. So encoding info in 2 char blocks is more efficient. I won't proceed with bigger block sizes because it doesn't provide further academic experience.

Footnotes:

[1] http://neerc.ifmo.ru/wiki/index.php?title=%D0%AD%D1%80%D0%B3%D0%BE%D0%B4%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%80%D0%BA%D0%BE%D

Author: Volkhov Mikhail, M3338
Created: 2016-11-04 Fri 14:06