陳志偉 B06901126 電機五

Electrical Engineering Lab（topics on Communication System）
Lab4 Report

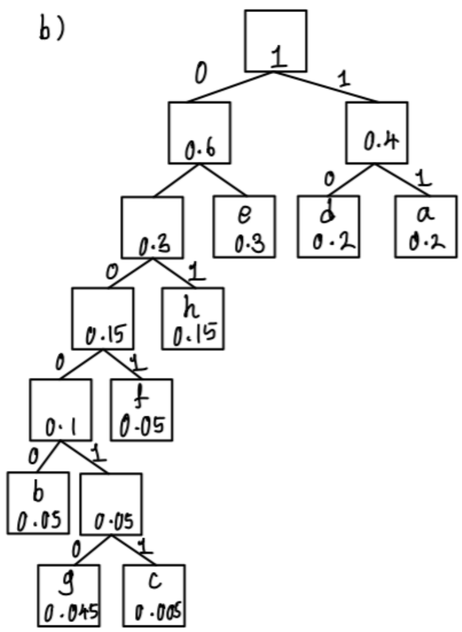
1. a) $H[X] = -\sum_{j} p_j \log p_j$

$= -[0.2\log 0.2 + 0.05\log 0.05 + 0.005\log 0.005 + 0.2\log 0.2 + 0.3\log 0.3 + 0.05\log 0.05$

$+ 0.045\log 0.045 + 0.15\log 0.15]$

$= 0.76225$

b)



Node	p	Left child	Right child	Bits
c	0.005			000011
g	0.045			000010
b	0.05			00000
f	0.05			0001
h	0.15			001
a	0.2			11
d	0.2			10
e	0.3			01
ad	0.4	d	a	
cg	0.05	g	c	
bcg	0.1	b	cg	
bcfg	0.15	bcg	f	
bcfgh	0.3	bcfg	h	
bcefgh	0.6	bcfgh	e	
abcdefgh	1	bcefgh	ad	

c) $\sum_{j=1}^{M} 2^{-l(a_j)} = 2^{-6} + 2^{-6} + 2^{-5} + 2^{-4} + 2^{-3} + 2^{-2} + 2^{-2} + 2^{-2}$

$= 1$

Satisfy the Kraft inequality

d) $\bar{L} = E[L] = \sum_{j=1}^{M} p(a_j)\, l(a_j) = 0.005\times 6 + 0.045\times 6 + 0.05\times 5 + 0.05\times 4 +$

$0.15\times 3 + 0.2\times 2 + 0.2\times 2 + 0.3\times 2$

$= 0.03 + 0.27 + 0.25 + 0.2 + 0.45 + 0.4 + 0.4 + 0.6$

$= 2.6$

e) {g, a, c, a, b}

↳ 000010 11 00011 11 00000

f) 000010 ⋮ 11 ⋮ 00011 ⋮ 11 ⋮ 00000

g ⋮ a ⋮ c ⋮ a ⋮ b

g) $T_\varepsilon^n = \{x^n : 2^{-n(H[X]+\varepsilon)} < p_{X^n}(x^n) < 2^{-n(H[X]-\varepsilon)}\}$

$= \{x^n : 0.0025 < p_{X^n}(x^n) < 0.01\}$

X : {c, abcdefghc, bcdefghc, ged, gea, bcgg, bcefghgbcg, begbcfgbcfgh, bh, fh}

2.a) Figure 1 show the dictionary of Table 1.

```
dict =

  8×2 cell array

    {'a'}    {[          1 1]}
    {'b'}    {[  0 0 0 0 0]}
    {'c'}    {[0 0 0 0 1 1]}
    {'d'}    {[          1 0]}
    {'e'}    {[          0 1]}
    {'f'}    {[      0 0 0 1]}
    {'g'}    {[0 0 0 0 1 0]}
    {'h'}    {[        0 0 1]}
```

Figure 1

2.b) After encoded the sequence of symbols {g, a, c, a, b}, the bits strings showed as Figure 2.

```
bin_seq =

     0     0     0     0     1     0     1     1     0     0     0     0     1     1     1     1     0     0     0     0     0
```

Figure 2

2.c) Let decode the bits strings from 2.b), we get the same result before we encoded the sequence of symbols {g, a, c, a, b}.

```
sym_seq =

  1×5 cell array

    {'g'}    {'a'}    {'c'}    {'a'}    {'b'}
```

Figure 3

3.a) From Figure 4, 'e' has the highest frequency showing in the random string which is satisfied with the probability in Table 1 and the length of the random string is 23 which is also satisfied with dictionary showed as Figure 1.

```
randomString =

    'edeebaddde'


data_length =

    23
```

Figure 4

3.b)

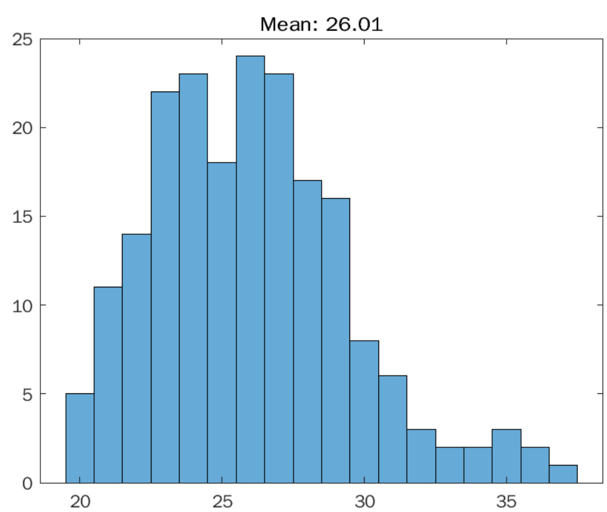


Figure 5

3.c)

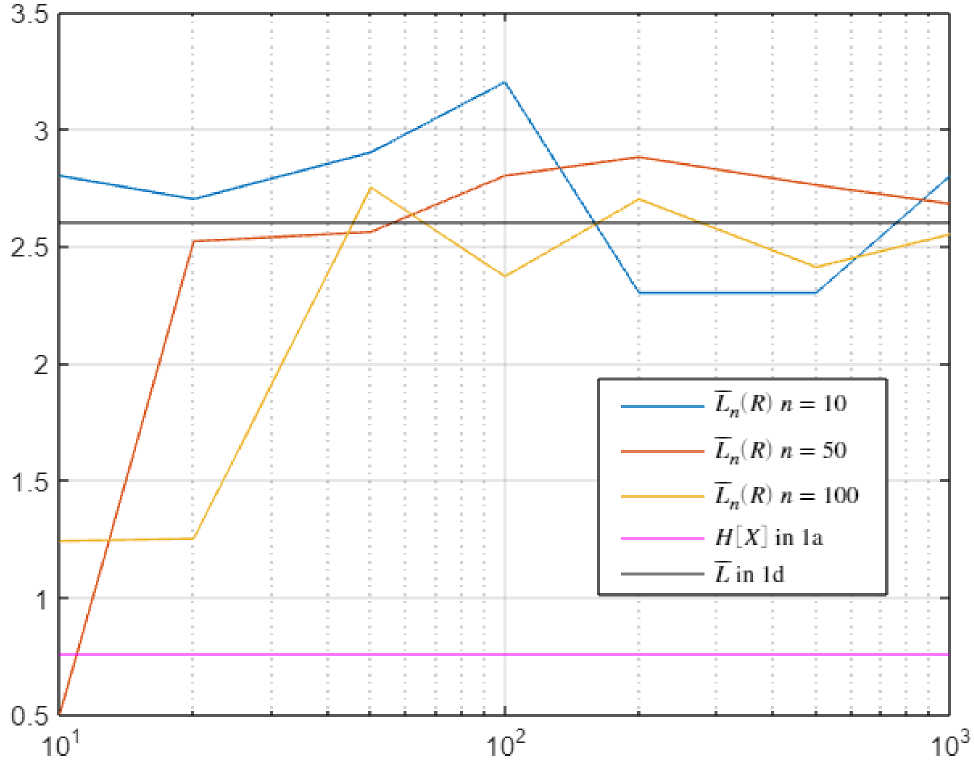


Figure 6

3.d) The three curves obtained in 3.c) will eventually converge to the answer we got in 1d, no matter how much N is equal to, as long as R is large enough. If N is large enough and R is small, the average codeword length will close to entropy.

Appendix

Code of problem 2

```
symbols = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
prob = [0.2, 0.05, 0.005, 0.2, 0.3, 0.05, 0.045, 0.15];

dict = huffmandict( symbols, prob );
display(dict);

sym_seq = {'g', 'a', 'c', 'a', 'b'};
display(sym_seq);

bin_seq = huffmanenco(sym_seq, dict);
display(bin_seq);

sym_seq = huffmandeco(bin_seq, dict);
display(sym_seq)
```

Code of problem 3

```
symbols = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
prob = [0.2, 0.05, 0.005, 0.2, 0.3, 0.05, 0.045, 0.15];

sym_bit = {'11', '00000', '000011', '10', '01', '0001', '000010', '001'};

N = [10, 50, 100];
R = [10 20 50 100 200 500 1000];
data_length_array = [];
avg_codeword = [];

for n = 1:length(N)
    for k = 1:length(R)
        for j = 1:R(k)
            indices = randsrc(N(n),1,[1:numel(symbols); prob]);
            randomString = [symbols{indices}];
            %display(randomString);
            data_length = 0;
            for i = 1:length(randomString)
                idx = find(strcmp(symbols, randomString(i)));
                data_length = data_length + length(sym_bit{idx});
            end
            data_length_array(k, j) = data_length;
        end
        mean = sum(data_length_array(n, k))/length(data_length_array(k));
        avg_codeword(n, k) = mean/N(n);
    end
end

entropy_1a = 0.76225;
avg_codeword_1d = 2.6;
```

```
display(avg_codeword(1,:))
semilogx(entropy_1a)
semilogx(R, avg_codeword)
yline(entropy_1a, Color='magenta')
yline(avg_codeword_1d, Color='black')
h = legend('$\overline{L}_n(R)$ $n=10$', '$\overline{L}_n(R)$ $n=50$', '$\overline{L}_n(R)$ $n=100$', '$H[X]$ in 1a', '$\overline{L}$ in 1d','Interpreter','latex');
rect = [0.6, 0.25, 0.25, 0.25];
set(h, 'Position', rect)
grid on
```

All code source will push to my github repo: https://github.com/finalwee/CommLab