Semaine 9

David Wiedemann

22 novembre 2020

1.1

L'entropie en bit est definie par

$$H(X) = p_1 \log_2 \frac{1}{p_1} + \ldots$$

Pour la distribution de probabilites dans cet exercice, on a

$$\begin{aligned}\frac{12}{24}\log_2 2 + \frac{8}{24}\log_2\frac{24}{8} + \frac{2}{24}\log_2\frac{24}{2} + \frac{2}{24}\log_2\frac{24}{2} \\ = 1 + \frac{8}{24}\log_2 3 + \frac{4}{24}\log_2 4 + \frac{4}{24}\log_2 3 \\ = \frac{4}{3} + \frac{1}{2}\log_2 3\end{aligned}$$

2.1.1

1.
La bande passante du signal est $611.2Hz$ et on echantillone avec une frequence de $\frac{1}{0.8\cdot 10^{-3}} = 1250Hz > 2 \cdot 611.2Hz$. Le signal peut donc etre parfaitement reconstruit.
2.
Si l'on echantillone avec une frequence de $\frac{1000}{1}Hz$, on a $611.2 \cdot 2 > 1000$, et donc le signal ne peut plus etre parfaitement reconstruit.

2.2.1

Voici une fonction, on suppose la libraire `math` importee.
On suppose que la suite de char est donnee dans un vector.

```
double entropie(vector<char> signal ) {
if ( signal.length( ) ==0){
        return 0;
}
```

```
double entropie=0;
bool pasfini= true;
char curchar= signal[0];
char firstchar=curchar;
size_t taille= signal.length( ) ;
while(pasfini) {
        double app=0;
        for(auto e: signal){
                        if(e == curchar){
                        app+=1;
                        }
        }
        entropie += app/taille * log(taille*1.0/app) / log ( 2.0)
        for( size_t i( 0);i<taille; ++i  ) {
                if(signal[i]!=curchar){
                signal[i]=curchar;
                }
        }

        pasfini=false;
        for( size_t i( 0); i<taille -1; ++i){
                if(signal[i]!=signal[i+1]){
                pasfini=true;
                }
        }
}
return entropie;
}
```

3.1

De manière générale, on a

$$H(X) = p_1 \log \frac{1}{p_1} + \dots$$

Donc

$$H(X) = \frac{n}{n+m}\log(\frac{n+m}{n}) + \frac{m}{n+m}\log(\frac{n+m}{m}) = p\log\frac{1}{p} + (1-p)\log\frac{1}{1-p} = h(p)$$

En effet, on a $1-p = \frac{n+m-n}{n+m}$.

3.2

En notant

$$H(A) = \sum_i \frac{a_i}{n} \log \frac{n}{a_i} \quad H(B) = \sum_i \frac{b_i}{n} \log \frac{n}{b_i}$$

On a

$$\begin{aligned} H(AB) &= \sum_i \frac{a_i}{n+m} \log \frac{n+m}{a_i} + \sum_j \frac{b_i}{n+m} \log \frac{n+m}{b_i} \\ &= p \sum_i \frac{a_i}{n} \log \frac{n+m}{a_i} + (1-p) \sum_j \frac{b_i}{m} \log \frac{n+m}{b_i} \\ &= p \sum_i \frac{a_i}{n} (\log \frac{1}{p} + \log \frac{n}{a_i}) + (1-p) \sum_j \frac{b_i}{m} (\log \frac{1}{1-p} + \log \frac{n+m}{b_i}) \end{aligned}$$

On remarque que la somme des a_i donne n, en développant donc la somme, on retrouve bien l'expression désirée :

$$H(AB) = h(p) + pH(A) + (1-p)H(B)$$

1.2

1.

Oui, c'est possible, l'entropie maximale est bornée par $\log_2 n$ où n est le nombre de lettres différentes, or $\log_2 33 \simeq 5.1$, le code de Huffman satisfait donc l'égalité

$$L(\text{ Huffman(X) }) - 1 = 4.5 \leq 5.1 \leq L(\text{ Huffman(X) }) = 5.5$$

2.

a)

La formule pour l'entropie est déjà donnée par

$$4.5 \leq H(X) \leq 5.5$$

La longueur moyenne d' un code de Shannon-Fano, noté $S(X)$ est donc

$$L(\text{ Huffman(X) }) \leq L_C(S(X)) \leq H(X) + 1$$

Donc

$$5.5 \leq L_C(S(X)) \leq 6.5$$

1.3

Le code 2 ne peut pas etre utilisé car il n'y pas de manière non-ambigue de représenter les niveaux 1 4 et 5 en utilisant seulement le rouge et le vert. C'est le seul code qui n'est pas utilisable.

2.2.2

On cherche à déterminer le taux de compression $T(X)$, On le notera sous-forme de pourcentages.
En utilisant le code de Huffman, on sait que le nombre moyen de questions sera compris entre 5.18 et 6.18.
Notons la longueur du code de Huffman L, on a alors

$$5.18 \leq L \leq 6.18$$

Soit n, le nombre de données(la longueur de la liste de toutes les valeurs)

$$5.18 \cdot n \leq L \cdot n \leq 6.18 \cdot n$$

Avant, la liste avait $8 \cdot n$ données, on a donc

$$\frac{5.18 \cdot n}{8 \cdot n} \leq \frac{L}{8} = T(X) \leq \frac{6.18 \cdot n}{8 \cdot n}$$

Donc, après simplification, on a

$$\frac{259}{400} \leq T(X) \leq \frac{309}{400}$$

2.2.3

On utilise le code de Huffman.
(Désolé, mais je suis pas pret à faire un arbre avec tikz, je vais simplement donner les différents codes binaires pour les différentes valeurs...)

Valeurs	Encodage
0	11
-32	10
32	010
-64	011
64	001
-127	0001
127	0000