ECOLE NATIONALE POLYTECHNIQUE

GENIE INDUSTRIEL

DS & AI

COMPTE RENDU DU TP1.

Compression de l'information sans pertes.

TINFO.

Fait par :

OUCHENE SOUHIL

1 DS & IA

I. PARTIE 1 :

➢ Question 0 :

On considère un alphabet de M symboles que nous notons x_i, avec 1≤ i ≤M. Chaque x_i symbole apparaît avec une probabilité P =P(x_i).

- Vérification de la loi de I(X_i) :

On a:

$$I(x_i)=Log_m(\frac{1}{P_i})=- Log_m(P_i)$$

D'après le support du cours : On sait que l'information I(x_i) apportée par la réalisation de l'évènement 'x_i' de probabilité P(x_i) est d'autant plus grande qu'il est improbable, ceci veut dire que

$$I(x_i)=f(\frac{1}{P(x_i)})$$

En se basant sur les caractéristiques du logarithme de base m, on pose $Log_m(x)=f(x)$

Car :

- Si P(x)→1 : évènement certain on aura aucune information ➔ I(X)→0.
- Si la probabilité de réalisation de X_i est P(X_i)=P_i alors $\frac{1}{P_i}$ est l'incertitude.
- Comme les deux évènements sont inversement proportionnel, il faut que la fonction *f* soit croissante.
- Si X_1 et X_2 sont indépendants : $f(X_1*X_2)= f(X_1) +f(X_2)$

- Vérification de la loi de H(X) :

On définit l'Entropie comme suit :

$$H(X)=-\sum_1^M Pi * logm(Pi)$$

On va maintenant calculer l'information totale, on aura :

$$I_{Totale} = \sum_{i=1}^{M} X_i * I(X_i)$$

On sait que ∀i=1 ;2 ; … ; M, on a : $P_i = \frac{Nombre\ d'occurrences\ des\ elements\ de\ realisation\ de\ Pi}{M}=\frac{Xi}{M}$

$$\Rightarrow X_i = P_i * M \rightarrow I_{Totale} = \sum_{i=1}^{M} Pi * M * I(X_i)$$

On va maintenant calculer l'information moyenne, on aura :

$$I_{moyenne} = \frac{1}{M} * I_{Totale}$$

$$\rightarrow I_{moyenne} = \frac{1}{M} * \sum_{i=1}^{M} Pi * M * I(X_i)$$

$$\rightarrow I_{moyenne} = \sum_{i=1}^{M} Pi * I(X_i)$$

Avec : $$I(x_i) = - Log_m(P_i)$$

$$\rightarrow I_{moyenne} = - \sum_{i=1}^{M} Pi * Log_m(P_i)$$

$$\rightarrow I_{moyenne} = H(X)$$

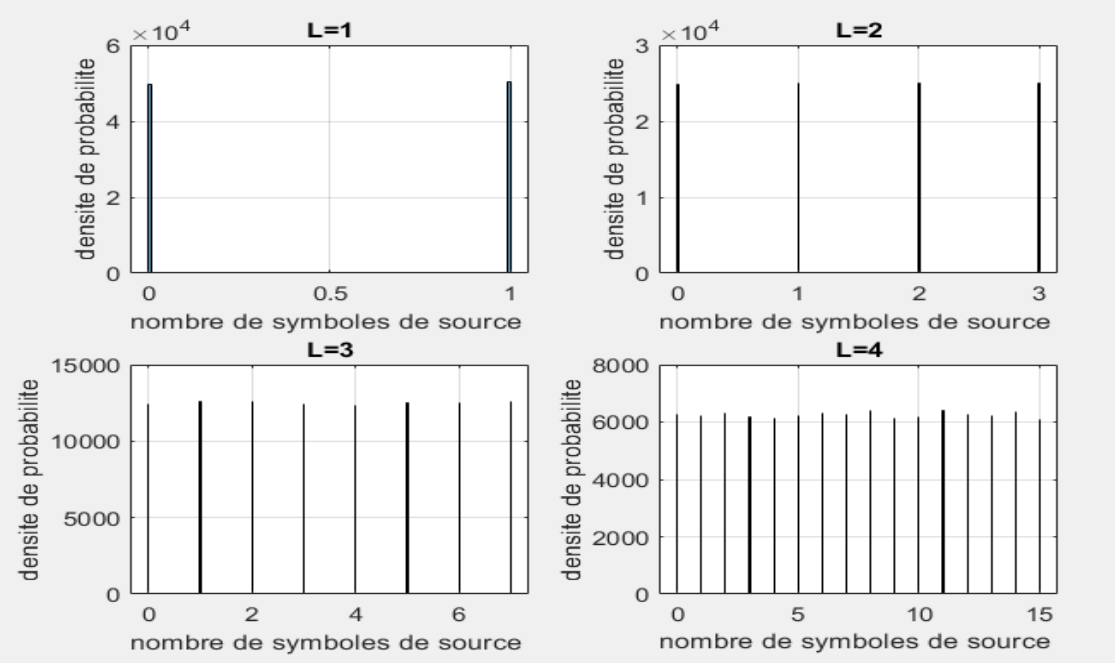
- ***Question 1 :***
  - ✓ ***CODE MATLAB :***

```
clc;
clf;
clear;
m=2;
N=100000;
L=1;
M=m^L;
X=randi([0 M-1],1,N);
h=0:0.01:M-1;
figure(1)
subplot(2,2,1);
hist(X,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('L=1');
grid;
L=2;
M=m^L;
X=randi([0 M-1],1,N);
h=0:0.01:M-1;
subplot(2,2,2);
hist(X,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('L=2');
grid;
L=3;
M=m^L;
X=randi([0 M-1],1,N);
h=0:0.01:M-1;
```

```
subplot(2,2,3);
hist(X,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('L=3');
grid;
L=4;
M=m^L;
X=randi([0 M-1],1,N);
h=0:0.01:M-1;
subplot(2,2,4);
hist(X,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('L=4');
grid;
```

PS : il était recommandé d'utiliser la fonction histogram au lieu de hist.

✓ *RESULTATS :*



✓ *REPONSES AUX QUESTIONS :*

- La fonction réalisée par hist(x,h) :

Tracer l'histogramme de la densité de probabilité (distribution de x) en fonction de nombre de symboles de source parmi les bins dont les centres sont spécifiés par h.

- La nature des symboles générés :

On constate que les symboles générés sont discrets, de distribution uniforme et équiprobable.

➢ *Question 2 :*

✓ *CODE MATLAB :*

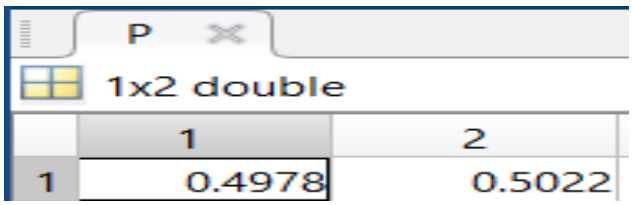
```
clc;clf;clear;m=2;N=100000;L=3;M=m^L;
X=randi([0 M-1],1,N);
h=0:0.01:M-1;
subplot(2,2,3);
histogram(X,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('L=3');
grid;
hist(X,h);
xlabel('nbre symbole de source ');
ylabel('DENSITE DE PROBABILITE');
title('');
grid;
frequence=hist(X,h);
Lf=length(frequence);
P=zeros(1,M);
I=zeros(1,M);
j=1;
for i=1:Lf
    if frequence(i)>0
        P(j)=frequence(i);
        j=j+1;

    end
end
P=P/N;
%calcul de l'information
I=-log2(P);
%calcul de l'entropie
H=0;
for i=1:M
    H=H-P(i)*log2(P(i))
End
H
```

✓ *REPONSES AU QUESTIONS :*

- Le contenu de P :

Pour L=1 :



Pour L=2 :

P

1x4 double

	1	2	3	4
1	0.2485	0.2514	0.2490	0.2510

Pour L=3 :

P

1x8 double

	1	2	3	4	5	6	7	8
1	0.1253	0.1255	0.1245	0.1240	0.1236	0.1260	0.1249	0.1262

Pour L=4 :

P

1x16 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.0624	0.0636	0.0622	0.0623	0.0619	0.0617	0.0621	0.0615	0.0630	0.0633	0.0641	0.0626	0.0617	0.0624	0.0629	0.0627

On peut remarquer que : ½ =0.5 ; ¼=0.25 ;1/8=0.125 et 1/16=0.0625

Les contenus de P sont proches à ces valeurs pour L_i : On peut dire que la distribution est équiprobable.

- Calcul de l'information et de l'entropie :

```
%calcul de l'information
I=-log2(P);
%calcul de l'entropie
H=0;
for i=1:M
    H=H-P(i)*log2(P(i))
End
H
```

- **Justification du choix de m=2 :**

On utilise un codage binaire, cad on considère le bit comme unité d'information dans le cas ou notre espace de probabilité est A= {0 ; 1}, on peut alors écrire :

$$I(x_i) = - \text{Log}_m(P_i) = - \frac{\log(p_i)}{\log(m)} = - C*\log(p_i)$$

On a aussi : $I(0) = - C*\log(1/2) = C * \log(2) = 1 \text{ bit}$

$C = \frac{1}{\log(2)}$: On utilise le logarithme de base 2

- Les valeurs de H :

Pour L=1 : H= 1.0000 :

```
I =

    1.0001    0.9999
```

Pour L=2 : H=2.0000 :

```
I =

    1.9964    1.9991    2.0070    1.9975
```

Pour L=3 : H= 2.9999 :

```
I =

    3.0131    3.0212    3.0003    2.9871    3.0017    2.9829    2.9909    3.0031
```

Pour L=4 : H= 3.9999 :

```
I =

    3.9892    3.9931    4.0065    4.0186    4.0414    4.0053    3.9977    3.9723    3.9956    4.0030    3.9965    3.9855    4.0121    3.9979    3.9780    4.0086
```

On remarque que H≈L dans les 4 cas, on sait que l'entropie soit max pour une source qui a des symboles qui apparient d'une manière équiprobable, et pour ces symboles équiprobables on n'aura aucune perte d'information ce qui est validé par la question 0 :

Pour $P_i = ½$ ➔ $I(x_i)=Log_2(2)= 1$ ∀i

Aussi, on voit que pour chaque L_i, P contient M=2^L éléments avec une probabilité

$$P_i = 1/M \rightarrow H(X)=\sum_1^M \frac{1}{M} * \log2(2\text{^}L)$$

$$=\sum_1^M \frac{1}{M} * L$$

$$=M * L/M = L$$

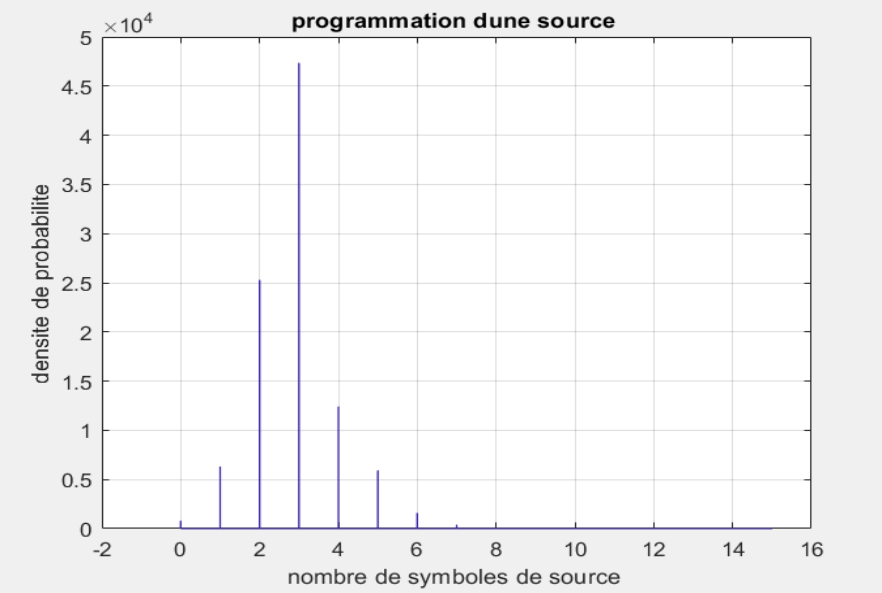
Ces 2 résultats confirment bien ce qu'on a obtenu en question 0.

- ***Question 3 :***
  - ✓ ***CODE MATLAB :***

```
M=8;
symbols=[0:M-1];
Pz=[0.0078 0.0625 0.2539 0.4713 0.1250 0.060 0.0156 0.0039];
z=randsrc(N,1,[symbols;Pz]);
hist(z,h);
xlabel('nombre de symboles de source ');
ylabel('densite de probabilite');
title('programmation dune source ');
grid;
```

```
frequence=hist(z,h);
Lf=length(frequence);
P=zeros(1,M);
Iz=zeros(1,M);
j=1;
for i=1:Lf
       if frequence(i)>0
        P(j)=frequence(i)/N;
        j=j+1;
    end
end
P;
Iz=-log2(P);
Hz=0;
for i=1:M
    Hz=Hz-P(i)*log2(P(i))
end
Hz
```

✓ ***RESULTATS :***



✓ ***REPONSES AUX QUESTIONS :***

- Comparaison entre P et Pz :

Valeurs de P :

```
P =

    0.0079    0.0613    0.2553    0.4682    0.1267    0.0605    0.0162    0.0039
```

Valeur de Pz :

```
Pz =

    0.0078    0.0625    0.2539    0.4713    0.1250    0.0600    0.0156    0.0039
```

Pour mieux voir la différence, on utilise le vecteur :

DIFF=P-Pz ; on aura alors :

```
DIFF =

    0.0001   -0.0012    0.0014   -0.0031    0.0017    0.0005    0.0006    0.0000
```

On constate que les symboles générés sont discrets et de distribution gaussienne (suivent la loi normale).

- Code pour Calculer de l'information et de l'entropie :

```
%calcul de l'information
Iz=-log2(P);
%calcul de l'entropie
Hz=0;
for i=1:M
    Hz=Hz-P(i)*log2(P(i))
end
Hz
```

- Comparaison entre H et Hz :

Calcul de Hz :

Il nous faut d'abord modifier le code un peut pour éliminer les valeurs nulles du vecteur P tel que :

```
P_not_null = P(find(P~=0));
```

Et puis on calcule le Hz comme indiqué dans le code ci-dessus, on aura alors :

L	1	2	3	4
H	1.0000	2.0000	2.9999	3.9999
Hz	0.0672	1.1903	2.0560	2.0570

On peut bien remarquer que L_i=H dans le 4 cas, ceci n'est pas vérifié pour Hz, et ceci est dû aux probabilités d'occurrence de chaque symbole, car on sait que l'entropie est maximale dans le cas équiprobable et sera égale a la longueur du code, ceci n'est pas vérifiée dans le cas des probabilités non égales, on en déduit que :

$$0 \leq H(x) \leq H_{max}$$

Tel que H_{max} est obtenue dans le cas equiprobable et sera egale a :

$$H_{max} = Log_2 (card(X))$$

II. PARTIE 2 :

✓ CODE MATLAB :

```
%creation du vecteur
symboles="On peut tout te prendre; tes biens, tes plus belles années,
l'ensemble de tes joies, et l'ensemble de tes mérites, jusqu'à ta dernière
chemise. Il te restera toujours tes rêves pour réinventer le monde que l'on
t'a confisqué.";
%nombre d'elements dans le vecteur
%on utilise soit la fonction strlength()
len=strlength(symboles);
%trouver les nombres d'occurrences pour chaque caractere: on utilise count
%on a considere que a=A ...
nb_a=count(symboles,'a','ignoreCase',true);
nb_b=count(symboles,'b','ignoreCase',true);
nb_c=count(symboles,'c','ignoreCase',true);
nb_d=count(symboles,'d','ignoreCase',true);
nb_e=count(symboles,'e','ignoreCase',true);
nb_f=count(symboles,'f','ignoreCase',true);
nb_g=count(symboles,'g','ignoreCase',true);
nb_h=count(symboles,'h','ignoreCase',true);
nb_i=count(symboles,'i','ignoreCase',true);
nb_j=count(symboles,'g','ignoreCase',true);
nb_k=count(symboles,'k','ignoreCase',true);
nb_l=count(symboles,'l','ignoreCase',true);
nb_m=count(symboles,'m','ignoreCase',true);
nb_n=count(symboles,'n','ignoreCase',true);
nb_o=count(symboles,'o','ignoreCase',true);
nb_p=count(symboles,'p','ignoreCase',true);
nb_q=count(symboles,'q','ignoreCase',true);
nb_r=count(symboles,'r','ignoreCase',true);
nb_s=count(symboles,'s','ignoreCase',true);
nb_t=count(symboles,'t','ignoreCase',true);
nb_u=count(symboles,'u','ignoreCase',true);
nb_v=count(symboles,'v','ignoreCase',true);
nb_w=count(symboles,'w','ignoreCase',true);
nb_x=count(symboles,'x','ignoreCase',true);
nb_y=count(symboles,'y','ignoreCase',true);
nb_z=count(symboles,'z','ignoreCase',true);
nb_point=count(symboles,'.','ignoreCase',true);
nb_vir=count(symboles,',','ignoreCase',true);
nb_e_accent_aigu=count(symboles,'é','ignoreCase',true);
nb_e_accent_grave=count(symboles,'è','ignoreCase',true);
nb_e_accent_circonflexe=count(symboles,'ê','ignoreCase',true);
nb_a_accent=count(symboles,'à','ignoreCase',true);
nb_semi_col=count(symboles,';','ignoreCase',true);
%pour les espaces on va utiliser la fonction : isstrprop(TEXT, type)
nb_spaces=0;
B = isstrprop(symboles, 'wspace');
for i=1:length(B)
    nb_spaces=nb_spaces+B(i);
end
nb_spaces;
%definition du vecteur de probabilites
P=1/len*[nb_a nb_b nb_c nb_d nb_e nb_f nb_g nb_h nb_i nb_g nb_k nb_l nb_m
nb_m nb_n nb_o nb_p nb_q nb_r nb_s nb_t nb_u nb_v nb_w nb_x nb_y nb_z
```

```
nb_point nb_vir nb_e_accent_aigu nb_a_accent nb_semi_col nb_e_accent_grave
nb_e_accent_circonflexe nb_spaces];
p_totale=0;
for i=1:length(P)
    p_totale=p_totale+P(i);
end
p_totale;
if p_totale<1
    disp('error, la somme des probabilites est <1');
end
%pour fixer ce probleme on peut ajouter une probabilite qui represente
les
%caracteres restants comme suit :
%nb_reste=1-p_totale;
%on peut aussi ajouter cette colonne au vecteur de probabilites comme
suit:
%P=1/len*[nb_a nb_b nb_c nb_d nb_e nb_f nb_g nb_h nb_i nb_g nb_k nb_l nb_m
nb_m nb_n nb_o nb_p nb_q nb_r nb_s nb_t nb_u nb_v nb_w nb_x nb_y nb_z
nb_point nb_vir nb_e_accent_aigu nb_a_accent nb_semi_col nb_e_accent_grave
nb_e_accent_circonflexe nb_spaces nb_reste*len];
%on peut revifier maintenant et ca va donner que la somme des Pi est 1
%p_totale=p_totale+P(length(P));
%if p_totale<1
 %   disp('error, la somme des probabilites est <1');
%elseif p_totale==1
 %   disp('everything is okey');
%end
%calcul de l'information propre:
Info_totale=-log2(p_totale);
Info_totale;
%on doit utiliser un tableau sans valeurs nulles pour qu'on puisse calculer

%l'entropie, alors on utilise la fonction find avec la condition P(i)<>0 et

%on stoque ces valeurs dans un vecteur P_not_null

P_not_null = P(find(P~=0));

%calcul de l'information propre par symbole:
Info=-log2(P_not_null)
Info
%calcul de l'entropie:
h=0;
for i=1:length(P_not_null)
   h=h-P_not_null(i)*log2(P_not_null(i));
end
h
%calcul de l'entropie dans le cas de symboles equiprobables
%cas de vecteur P ayant des probabilites nulles :
h_equiprobable_null=log2(length(P))
h_equiprobable_null
%cas de vecteur P ayant des probabilites nulles :
h_equiprobable=log2(length(P_not_null))
h_equiprobable
%mesure de l'efficacite
efficacite = h/h_equiprobable;
efficacite
%efficacite en %
```

```
efficacite*100
%mesure de la redondance:
%  on utilise 1-efficacite
redondance=1-efficacite;
redondance*100
```

✓ *REPONSES AUX QUESTIONS :*

- **Calcul de l'information propre:**

Pour l'information totale, on a **: Info_totale = 0.0194**
Pour l'information par symbole, on a :
Info =

5.8138	**5.8138**	**6.8138**	**5.4919**	**2.6043**	**7.8138**	**7.8138**	**4.8138**	**4.4919**	**5.4919**
5.4919	**4.1133**	**4.6439**	**5.8138**	**6.2288**	**4.2288**	**3.5659**	**3.7263**	**4.4919**	**6.8138**
6.8138	**5.8138**	**5.8138**	**7.8138**	**7.8138**	**7.8138**	**7.8138**	**2.6043**		

- **Calcul de l'entropie :**

h =4.0146

- **Nombre de bits nécessaires :**

Dans le cas de symboles équiprobables, l'entropie sera max et sera égale à :

h_equiprobable=H_{max}=$\log_2$(length(P_not_null))

On aura alors **: H_{max}= h_equiprobable = 4.8074**
Cad : on prend **5 bits par symboles** pour coder le message dans le cas équiprobable.

- **Oui,** il est possible d'avoir un nombre inférieur, car on peut prendre par exemple le codage de 'e' qui apparait 37 fois pour seulement 2 ou 3 bits, et prendre 5 ou 6 bits pour le codage de 'c' qui apparait seulement 2 fois dans le message, ceci sera bénéfique pour la mémoire :
  Coder les symboles plus probables sur un nombre minimal de bits et vice versa pour les symboles peu probables.

- **Calcul d'efficacité et de redondance :**

Pour l'efficacité, on a : **efficacite = 0.8351 ➔ efficacite (en %) = 83.5096 %**
Pour la redondance : **redondance=1-efficacite = 0.1649 ➔ redondance (en %) = 16.4904%**