Ecole Internationale des Sciences de Traitement de l'Information

Introduction à la théorie d'Information

Travaux dirigés.

Anya Désilles

Année scolaire 2010/2011

EISTI. **Année scolaire 2010-2011**

Département "Informatique"

Théorie de l'information

Série d'exercices N°1

Partie I. Entropie d'une source. Définitions et propriétés.

Exercice 1 (Calcul d'entropie.)**.** Soit une source d'alphabet $\Omega = \{1, 2, 3, 5, 4\}$. Calculer son entropie pour les distributions de probabilités suivantes.

1. $P_1 = \{0.2, 0.2, 0.2, 0.2, 0.2\}$
2. $P_2 = \{0.05, 0.05, 0.05, 0.05, 0.8\}$
3. $P_3 = \{0.1, 0.2, 0.3, 0.15, 0.25\}$

Exercice 2.

1. On lance une pièce dont les deux cotés sont identiques : pile. Quelle est l'entropie associée à cette expérience ?
2. On lance un dé équilibré à 6 faces. Quelle est l'information moyenne apportée par l'observation de la parité du résultat ?
3. Un jeu de cartes contient 3 piques, 4 trèfles, 2 cœurs et 1 carreau. On tire une carte au hasard. Quelle est l'entropie de l'observation de la couleur de la carte ?

Exercice 3 (Propriété de groupe.)**.** L'objectif de cet exercice est de vérifier sur un exemple une propriété importante de la fonction d'entropie. Soit X une source d'alphabet $\Omega_X = \{x_1, \ldots, x_n\}$ et de distribution de probabilités $P_X = \{p_1, \ldots, p_n\}$. Soit $1 \leq r < n$. On divise l'alphabet en deux sois-ensembles $A = \{x_1, \ldots, x_r\}$ et $B = \{x_{r+1}, \ldots, x_n\}$ de telle sorte que $\Omega_X = A \cup B$ et $A \cap B = \emptyset$. Soit $p = P(A)$ et $q = 1 - p = P(B)$. Alors on a la relation suivante :

$$H(X) = H(p_1, \ldots, p_n) = H(p, 1-p) + pH\left(\frac{p_1}{p}, \ldots, \frac{p_r}{p}\right) + (1-p)H\left(\frac{p_{r+1}}{1-p}, \ldots, \frac{p_n}{1-p}\right)$$

On remarquera que $P_A = \left\{\frac{p_1}{p}, \ldots, \frac{p_r}{p}\right\}$ et $Q_B = \left\{\frac{p_{r+1}}{1-p}, \ldots, \frac{p_n}{1-p}\right\}$ définissent des distributions de probabilité respectivement sur A et B.

Soit X une source d'alphabet $\Omega = \{1, 2, 3, 5, 4\}$ de distribution de probabilité $P = \{0.1, 0.2, 0.3, 0.15, 0.25\}$. Soient $A = \{1, 3, 5\}$ et $B = \{2, 4\}$.

1. Calculer l'entropie de X.
2. Calculer $p = P(A)$ et $q = P(B)$. En déduire $H(p, q)$.
3. Définir les distributions de probabilités sur A et sur B comme indiqué ci-dessus et les entropies associées. Vérifier ensuite la propriété de groupe.

Exercice 4 (Vers le codage de Shannon.)**.** Reprenons la même source que dans l'exercice précédent X d'alphabet $\Omega = \{1, 2, 3, 5, 4\}$ de distribution de probabilité $P = \{0.1, 0.2, 0.3, 0.15, 0.25\}$. Supposons que l'on doit deviner le symbole émis par la source et que l'on a droit de poser des questions binaires (réponses possibles "oui" et "non"). On cherche à construire la stratégie qui, en moyenne, permet de trouver la réponse en un nombre minimal de questions.

Remarquons qu'une question binaire induit sur l'ensemble Ω une partition en deux sous-ensembles A et B correspondants aux réponses "oui" et "non". Par exemple, si l'on demande "est que le nombre est pair ?" la partition sera celle de l'exercice précédent. Soit $p = P(A)$ la probabilité de la réponse "oui" à une question donnée.

1. Quelle est l'information moyenne obtenue par la réponse à une question binaire ?
2. Quelle est l'information moyenne maximale ? Et pour quelle valeur de p est est atteinte ?
3. Quel est alors le meilleur choix de première question à poser ?
4. Appliquer le même raisonnement récursivement pour choisir la meilleure deuxième question selon la réponse à la première. Continuer jusqu'à arriver à identifier chaque symbole.
5. Construire un arbre représentant la stratégie obtenue.
6. Calculer le nombre moyen de questions. Comparer à l'entropie de X.

EISTI. **Année scolaire 2010-2011**

Département "Informatique"

Théorie de l'information

Série d'exercices N^{o}2

Partie I. Entropie : un jeu d'espion !

Le chiffrement de César consiste à décaler l'alphabet de k positions de façon cyclique et de remplacer chaque lettre d'un massage clair par une lettre correspondante de l'alphabet décalé. La clé secrète de ce chiffre est un entier $1 \leq k \leq 25$ qui représente le décalage de l'alphabet. Nous allons dans un premier temps étudier la sécurité de ce chiffre et ensuite apprendre la méthode d'analyse des fréquences qui a permis de la casser.

Exercice 5 (Rendons à César ce qui est à César : son chiffre !)**.**

1. Montrer que pour les messages de longueur $l = 1$ le chiffre de César est parfaitement sûr au sens de Shannon. Pour cela montrez que

$$\forall m \in \mathcal{M}, \quad \forall c \in \mathcal{C}, \quad P[M = m|C = c] = P[M = m]$$

2. Montrer que pour les messages de longueur $l \geq 2$ le chiffre n'est plus parfaitement sûr. Pour cela analysez l'exemple suivant. Soient le message $m = AB$ et le chiffré $c = DM$. Montrer que $P[M = m|C = c] = 0$ tandis que $P[M = m] \neq 0$.

Exercice 6 (Cryptanalyse du chiffre de César)**.** La méthode d'analyse des fréquences a été inventé par le savant arabe AL-Kindi au IX-ème siècle. On suppose que l'on connaît la langue du texte clair et que l'on dispose du message chiffré. Dans le ca de chiffre de César on cherche à déterminer le paramètre k, clé du chiffre. La méthode consiste à comparer l'histogramme d'occurrences des caractères du chiffré avec la table des fréquences d'occurrence des caractères de la langue du texte clair. Voici la table des fréquences de la langue française.

Lettre	Fréquence %	Lettre	Fréquence %
A	8.4	N	7.13
B	1.06	O	5.26
C	3.03	P	3.01
D	4.18	Q	0.99
E	17.26	R	6.55
F	1.12	S	8.08
G	1.27	T	7.07
H	0.92	U	5.74
I	7.34	V	1.32
J	0.31	W	0.04
K	0.05	X	0.45
L	6.01	Y	0.3
M	2.96	Z	0.12

1. Votre mission, si vous l'acceptez, consiste à déchiffrer le message secret de votre binôme. Chacun de vous va composer un message de son choix. Pour le chiffrer, utiliser `http://www.bibmath.net/crypto/substi/cryptcesar.php3`l'applet java sur ce site. Envoyez le chiffré obtenu à votre voisin (par e-mail ou chat). Ensuite chacun cherchera à trouver la clé et le message clair par analyse fréquentielle.
2. Et maintenant, déchiffrez ceci :

UHGCHNK. GHNL OHNL IKHIHLHGL MKHBL PTZHGL IHNK ITKMBK TN STGBSBUTK. OHMKX TOBHG OT ITKMBK ET HN OHNL OHNEXS. NG OKTB OTNMHNK OXNM MHNCHNKL OHEXK ATNM IHNK OHBK LT IKHBX. NG SHFUB FTKVATBM XG SBZSTZTGM LNK ET KHNMX.

Partie II. Entropie et arbres de décision

Exercice 7 (Le jeu de la fausse pièce)**.** On considère un jeu de n pièces, toutes d'apparence identiques. On sait qu'une seule pièce est fausse. Elle a un poids différent des autres mais on ne sait pas si elle plus légère ou plus lourde.

On dispose d'une balance à deux plateaux. À chaque pesée la balance peut se trouver dans l'une des trois configurations :

G Elle penche vers la gauche.

D Elle penche vers la droite.

E Elle reste à l'équilibre.

L'objectif est de déterminer avec le moins de pesées possible la fausse pièce et si elle plus légère ou plus lourde.

1. Soit $n = 8$.
   (a) Combien de réponses possibles il y a dans ce problème ?
   (b) Combien de possibilités différentes peut on obtenir avec 2 pesées ? Avec 3 pesées ? Conclusion sur la faisabilité de détermination de la fausse pièce en 2 ou en 3 pesées.
   (c) Est-il intéressant de peser deux lots de 4 pièces chacun ? Pourquoi ? Combien d'information nous apporterait une telle pesée ?
   (d) A l'aide du programme java à l'adresse `http://nlvm.usu.edu/fr/nav/frames_asid_139_g_4_t_2.html` Élaborer une stratégie de pesée permettant de déterminer la fausse pièce en 3 pesées dans tous les cas. Voici quelques conseils qui peuvent être utiles
      i. Essayez, pour commencer, de diviser l'ensemble de vos pièces en trois lots, de taille à peu près égale. Pour 8 pièces vous pouvez tester les décompositions $8 = 3+3+2$ ou $8 = 2+2+4$;
      ii. Lorsque l'on compare le même lot de pièces $L1$ à deux lots différents $L2$ et $L3$ on peut interpréter les résultats de façon suivante : si les deux résultats de comparaison sont identiques, la fausse pièce est dans le lot 1 et nous avons le sens de la différence de poids ($+$ ou $-$) ; il n'est pas possible que l'un des résultats soit G et l'autre D.

(e) Essayez de représenter votre stratégie sous forme d'arbre de décision. Les feuilles de l'arbre doivent représenter toutes les réponses possibles. La profondeur de cet arbre indique alors le nombre maximal de pesées.

2. A l'aide du même programme java élaborer une stratégie pour $n = 9, 10, 12$.

EISTI. **Année scolaire 2010-2011**

Département ” Informatique ”

Théorie de l'information

Série d'exercices N^{o}3

Partie I. Codes sans préfixe.

Nous allons considérer une source d'alphabet

$$\Omega = \{a, b, c, d, e, f, g, h, i, j\}$$

et un canal d'alphabet binaire $\{0, 1\}$.

Nous utiliserons dans la suite la terminologie suivante :

Lettre, symbole ou caractère Tout élément d'un alphabet donné ;

Message ou mot Une séquence fini m de caractères d'un alphabet donné ;

Longueur de mot le nombre $l(m)$ de caractères d'un mot m ;

Voici les définitions importantes.

Code. Un code binaire pour l'alphabet donné Ω de taille n est un ensemble de n mots binaires $\{m_1, \ldots, m_n\}$. Un code est régulier si les mots correspondants aux différents caractères de l'alphabet sont différents. Dans la suite nous étudions uniquement les codes réguliers.

Exemple. Pour notre alphabet un code régulier peut être

s_i	a	b	c	d	e	f	g	h	i	j
m_i	0	1	00	01	11	10	100	101	110	111

Code déchiffrable Un code binaire est déchiffrable si toute séquence de bits peut être décodée de façon unique.

Exemple. Le code ci-dessus n'est pas déchiffrable. En effet, la séquence de bits ”110” peut être interprétée comme ”i” ou comme ”bf” ou même comme ”ea”.

Code sans préfixe. Un code est sans préfixe ou instantané si aucun mot code n'est le préfixe d'un autre. Le code ci-dessus n'est pas un code sans préfixe. En effet, le mot code 0 est le début des mots du code 00, 01. Le code suivant est sans préfixe :

s_i	a	b	c	d	e	f	g	h	i	j
m_i	1111	1110	110	1011	1010	100	011	000	010	001

Décodage pas à pas d'un code sans préfixe. Le principe de décodage est simple. Il suffit de lire la séquence codée de gauche à droite jusqu'à ce qu'on trouve un mot du code. On est alors certain que ce mot correspond sans ambiguïté à un seul caractère de l'alphabet. On enregistre le caractère et on recommence la lecture.

Exercice 8. Appliquez la procédure de décodage pas à pas à la séquence 10110000101010100.

Théorème 0.1 (Inégalité de Kraft)**.** *Un code instantané de longueurs de mots données* $l_1, \ldots, l_n$ *existe si et seulement si*

$$\sum_{i=1}^{n} d^{-l_i} \leq 1$$

où d *est la taille de l'alphabet du canal.*

Exercice 9. Vérifier s'il existe un code sans préfixe de longueurs de mots données : $\{4, 4, 4, 3, 2, 4, 3, 4, 3, 4\}$.

Partie II. Codes sans préfixe et arbres binaires.

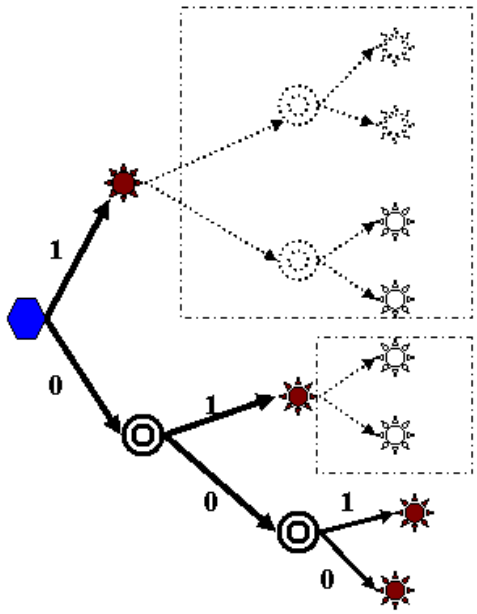


FIGURE 1 – Exemple

Vous trouverez dans le polycopié du cours les rappels nécessaires de vocabulaire associé aux arbres binaires. (voir page 44).

Soit $\{m_1, \ldots, m_n\}$ un code binaire sans préfixes de longueur maximale l. Il est évident que chaque mot de ce code peut être représenté par un chemin partant de la racine d'un arbre binaire complet de profondeur l. il suffit pour cela d'étiqueter les arcs du l'arbre avec 0 et 1. Supposons qu'à un mot m_i de longueur $l_i \leq l$ on vient d'associer un chemin de l_i arcs en partant de la racine. Le chemin s'arrête alors à un noeud de niveau l_i. Comme aucun autre mot du code ne peut avoir celui ci comme préfixe, on peut supprimer tous les descendants du noeud final du

chemin. Ce dernier devient alors une feuille. Ainsi on peut associer à tout code sans préfixes un arbre binaire incomplet. L'arbre vu précédemment, est ainsi associé au code $\{1, 01, 001, 000\}$ (voir la figure 1).

Ainsi dans un arbre correspondant à un code binaire, les feuilles correspondent aux mots du code. Si ce dernier est associé à l'alphabet d'une source, il est également possible d'associer aux feuilles les probabilités des symboles correspondants.

Exercice 10.

1. Construire l'arbre associé au code ci-dessous

s_i	a	b	c	d	e	f	g	h	i	j
m_i	1111	1110	110	1011	1010	100	011	000	010	001

2. Construire le code sans préfixe de longueurs de mots données dans l'exercice 2 à l'aide d'un arbre binaire, en procédant à l'élagage d'un arbre complet.

Partie III. Codage de Huffman

Nous allons considérer une source d'alphabet

$$\Omega = \{a, b, c, d, e, f, g, h, i, j\}$$

et un canal d'alphabet binaire $\{0, 1\}$. Voici les définitions importantes.

Théorème de la borne inférieure de longueur de code

Théorème 0.2. *Soit une source S d'alphabet $\Omega_S = \{s_1, \ldots, s_n\}$ de taille n et de distribution de probabilités $P_S = \{p_1, \ldots, p_n\}$. Soit un canal d'alphabet binaire $\Omega_C = \{0, 1\}$ sans bruit, stationnaire et sans mémoire. Soit un code déchiffrable $\{m_1, \ldots, m_n\}$ de longueurs de mots $\{l_1, \ldots, l_n\}$.*
Alors la longueur moyenne de mots de code vérifie :

$$\bar{L} = \sum_{i=1}^{n} p(s_i) l_i \geq H(S)$$

L'égalité n'est possible que si $\forall i = 1, \ldots, n, \quad p_i = 2^{-l_i}$.

Code absolument optimal C'est un code dont la longueur moyenne de mots est égale à la borne inférieure, $H(S)$.

Code optimal. Un code est dit optimal dans une certaine classe de codes si sa longueur moyenne de mots est minimale dans cette classe. La classe de codes la plus importante est celle de codes sans préfixe.

Exercice 11. Soit une source d'alphabet Ω et de distribution de probabilité suivante

s_i	a	b	c	d	e	f	g	h	i	j
m_i	0.2	0.05	0.1	0.05	0.15	0.05	0.1	0.05	0.15	0.1

1. Quelle est la longueur moyenne minimale pour un code binaire de cette source ?
2. Existe-t-il un code absolument optimal ?
3. Construire un code sans préfixe optimal selon la méthode de Huffman.
4. Représenter ce code sous forme d'arbre.

EISTI. **Année scolaire 2010-2011**

Département ” Informatique ”

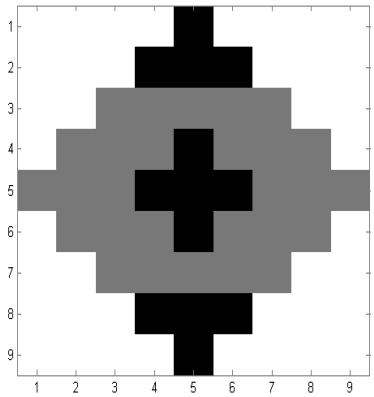
Théorie de l’information

Série d’exercices N^{o}4

Codage et compression d’images sans pertes

Dans ce TD nous allons étudier quelques applications de techniques de codage à la compression sans pertes des images. Nous allons expérimenter sur la même image plusieurs approches de compression et en choisir la meilleure.

Voici l’image que nous allons analyser :



Elle est de taille 9×9 pixels. On suppose que chaque pixel de l’image est encodé sur 8 bits (1 Octet) ce qui donne une image en niveaux de gris avec une palette de 256 niveaux possibles.

Or, notre image ne contient trois couleurs : Noir (N=256), Gris (G=128), Blanc (B=0). Si on représente chaque pixel par le caractère qui représente sa couleur on obtient la matrice

$$\begin{bmatrix} B & B & B & B & N & B & B & B & B \\ B & B & B & N & N & N & B & B & B \\ B & B & G & G & G & G & G & B & B \\ B & G & G & G & N & G & G & G & B \\ G & G & G & N & N & N & G & G & G \\ B & G & G & G & N & G & G & G & B \\ B & B & G & G & G & G & G & B & B \\ B & B & B & N & N & N & B & B & B \\ B & B & B & B & N & B & B & B & B \end{bmatrix}$$

On notera pour référence que si chaque pixel est codé par 8 bits, l’image occupe

$$9 \times 9 \times 8 = 648 \text{ bits}$$

ou encore 81 Octets.

Exercice 12 (Codage de Huffman)**.** Dans cette exercice nous allons appliquer à l'image le codage de Huffman en considérant les couleurs comme l'alphabet de la source. Pour toute lecture, l'image sera parcourue "ligne par ligne" en commençant par le coin en haut à gauche.

1. Constituez la table des fréquences des couleurs **présentes** dans l'image.
2. Calculez l'entropie associée. Que dire de l'efficacité du codage qui associe 8 bits à chaque pixel. Peut on trouver un codage plus économique pour cette image ?
3. A partir de votre table des fréquences, calculez l'arbre de Huffman et le code associé.
4. Calculez en bits la taille du code de Huffman pour cette image. Quel est le taux de compression ?

Exercice 13 (Codage RLE)**.** Nous allons maintenant appliquer une autre technique à la même image : le codage RLE. Cette méthode consiste à remplacer chaque séquence de caractères identiques par le couple (nb, c) où nb est le nombre de caractères de la séquence et c est le caractère répété. Par exemple, la séquence de couleurs $NNNBBBGGG$ sera codée par

$$(3, N)(3, B)(3, G)$$

1. En parcourant l'image ligne (sans interruptions pour les sauts de ligne !) constituez le code RLE correspondant. Commencez le codage en haut à gauche.
2. Chaque couple obtenu est encodé sur 16 bits : 8 pour le nombre et 8 pour le caractère. Quelle est la longueur en bits du code obtenu ?
3. Calculer le taux de compression.

Exercice 14 (RLE+Huffman)**.** On reprend le code RLE obtenu dans l'exercice précédent. Certains couples (nb, C) se répètent et le nombre de couples réellement présents dans le code n'est pas élevé. On va utiliser le code de Huffman pour réduire la place en mémoire de chaque couple.

1. Constituez la table des couples (nb, C) **présents** dans le code RLE de l'image.
2. Calculez l'entropie associée. Que dire de l'efficacité du codage qui associe 16 bits à chaque couple. Peut on trouver un codage plus économique ?
3. A partir de votre table des fréquences, calculez l'arbre de Huffman et le code associé.
4. Calculez en bits la taille du code de Huffman pour la séquence du code RLE de l'image. Quel est le taux de compression ?
5. Comparez les performances des trois méthodes.

EISTI. **Année scolaire 2010-2011**

Département " Informatique "

Théorie de l'information

Série d'exercices N^{o}5

Partie I. Représentation mathématique d'un canal de transmission

---

Exercice 15 (Exemple)**.** Soient X et Y deux variables aléatoires prenant leurs valeurs respectivement dans $\Omega_X = \{x_1, x_2, x_3\}$ et $\Omega_Y = \{y_1, y_2\}$ et ayant la matrice de probabilités conjointes suivante

$$P(X,Y) = \begin{array}{c|cc} & y_1 & y_2 \\ \hline x_1 & 0.25 & 0 \\ x_2 & 0.1 & 0.3 \\ x_3 & 0.1 & 0.25 \end{array}$$

En déduire les distributions conditionnelles et marginales.

Solution de l'exercice 19

Distributions marginales de X et Y. On utilise les définitions

$$\forall i = 1, \ldots, 3,\ p(x_i) = \sum_{j=1}^{2} p(x_i, y_j), \quad \forall j = 1, \ldots, 2,\ p(y_j) = \sum_{i=1}^{3} p(x_i, y_j)$$

On peut résumer ces calculs sous forme d'un tableau

	y_1	y_2	P_X
x_1	0.25	0	0.25
x_2	0.1	0.3	0.4
x_3	0.1	0.25	0.35
P_Y	0.45	0.55	

La dernière colonne de ce tableau représente la distribution marginale de X et s'obtient en calculant les sommes des éléments de chaque ligne.

La dernière ligne du tableau représente la distribution marginale de Y et s'obtient en calculant les sommes des éléments de chaque colonne.

Distributions conditionnelles $P(X|Y)$ et $P(Y|X)$. On utilise la définition

$$\forall i = 1, \ldots, 3,\ \forall j = 1, \ldots, 2,\ p(x_i|y_j) = \frac{p(x_i, y_j)}{p(y_j)}, \quad p(y_j|x_i) = \frac{p(x_i, y_j)}{p(x_i)}$$

On trouve les matrices

$$P(X|Y) = \begin{pmatrix} 5/9 & 0 \\ 2/9 & 6/11 \\ 2/9 & 5/11 \end{pmatrix}, \qquad P(Y|X) = \begin{pmatrix} 1 & 0 \\ 1/4 & 3/4 \\ 2/7 & 5/7 \end{pmatrix}$$

Exercice 16 (Un modèle probabiliste de transmission)**.** Soit une source binaire qui émet des signaux binaires, composés de 0 et de 1. On associé à l'expérience d'envoi d'un seul symbole une variable aléatoire X qui prend donc des valeurs dans $\Omega_X = \{0, 1\}$. On suppose que la source émet 0 ou 1 avec équiprobabilité. Autrement dit, la distribution de X est connue :

$$p_X(0) = p_X(1) = 0.5$$

Les symboles émis sont ensuite envoyés via un canal de transmission ayant des perturbations aléatoires. On associe à l'expérience d'observation du symbole reçu la variable aléatoire Y qui peut prendre trois valeurs : $\Omega_Y = \{-1, 0, 1\}$. La valeur -1 correspond au cas où le système n'est pas capable d'identifier un 0 ou un 1 à la sortie. On a établi les probabilités de réception suivantes :

- Si le symbole envoyé X est 0 alors on reçoit :
  1. $Y = 1$ avec la probabilité 0.2
  2. $Y = 0$ avec la probabilité 0.7
  3. $Y = -1$ avec la probabilité 0.1
- Si le symbole envoyé X est 1 alors on reçoit :
  1. $Y = 1$ avec la probabilité 0.6
  2. $Y = 0$ avec la probabilité 0.3
  3. $Y = -1$ avec la probabilité 0.1

1. Dire si les probabilités de réception données ci-dessus définissent la distribution conjointe ou conditionnelle de X et Y ? Si conditionnelle, préciser de quelle variable ?
2. Pouvez-vous donner une représentation sous forme de graphe des probabilités de réception ?
3. Former une matrice à partir des probabilités de réception.
4. Calculer toutes les distributions manquantes.
5. Calculer les entropies associées au canal :

$$H(X),\ H(Y),\ H(X,Y),\ H(X|Y),\ H(Y|X)$$

6. En déduire l'information mutuelle $I(X, Y)$ du canal.

Exercice 17. Soit une source binaire X d'alphabet $\Omega_X = \{0, 1\}$. On suppose que la distribution de probabilité de X est connue :

$$p_X(0) = p, \quad p_X(1) = 1 - p$$

Les symboles émis sont envoyés via un canal de transmission ayant des perturbations aléatoires. On associe à l'expérience d'observation du symbole reçu la variable aléatoire Y qui peut prendre trois valeurs : $\Omega_Y = \{-1, 0, 1\}$. La valeur -1 correspond au cas où le système n'est pas capable d'identifier un 0 ou un 1 à la sortie. On suppose que la matrice de transition du canal est la suivante :

$P(Y|X) =$

$X \backslash Y$	0	-1	1
0	0.8	0.2	0
1	0	0.2	0.8

1. Est ce que c'est un canal symétrique ? Justifier.
2. En utilisant la matrice de transition $P(Y|X)$ et la distribution de probabilité de X, P_X, calculer la distribution de probabilité conjointe $P(X, Y)$. En déduire P_Y, la distribution marginale du récepteur, Y et la distribution conditionnelle $P(X|Y)$.
3. Exprimer l'entropie $H_X(p)$ comme fonction du paramètre p.
4. Calculer $H(X|Y)$ en fonction du paramètre p.
5. Calculer $I(X;Y) = H_X(p) - H(X|Y)$. Montrer que $I(X;Y) = 0.8H_X(p)$.
6. En déduire la valeur de la capacité du canal définie par

$$C = \max_{P_X} I(X;Y)$$

où le maximum est pris selon toutes les distributions possibles de la source X.

Partie III. Pour approfondir

Exercice 18. Soit l'algorithme suivant.

Entrée : $x_0 \in \{1, 2, 3, 4\}$ tiré aléatoirement avec équiprobabilité.

Algorithme : $n \leftarrow 0$

Tant que $(x_n \neq 1)$

$$A_n \leftarrow \begin{cases} 0, \text{ avec probabilité } & p_n = 1/x_n \\ 1, \text{ avec probabilité } & q_n = 1 - 1/x_n \end{cases}$$

Si (x_n est pair)

Alors $x_{n+1} \leftarrow \dfrac{x_n}{2}$

Sinon $x_{n+1} \leftarrow A_n x_n + 1$

Fin Si

$n \leftarrow n + 1$

Fin Tant que

Sortie : Nombre d'itérations n.

Soient X, la variable aléatoire correspondante à l'entrée de l'algorithme, uniformément distribuée sur $\{1, 2, 3, 4, \}$ et N la variable aléatoire correspondante à la sortie de l'algorithme.

L'objectif de l'exercice *est de calculer le nombre moyen d'itérations effectuées par cet algorithme. Autrement dit, calculer* $E[N]$.

Voici le plan à suivre.

1. Etablir la matrice de probabilités conditionnelles $P(N|X)$, en utilisant les schémas sous forme d'arbre pour le déroulement de l'algorithme pour chaque valeur de X.Voir l'exemple sur la figure ci-dessous.
2. Sachant que X est uniformément distribué, déduire la matrice de probabilités conjointes de N et X.
3. Calculer la distribution de probabilités marginale de N
4. En déduire le nombre moyen d'itérations.

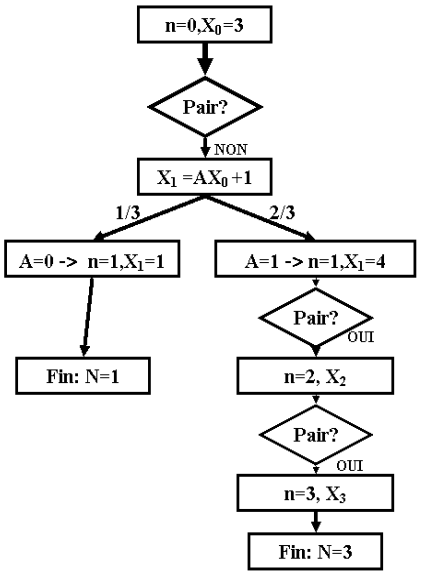


FIGURE 2 – $X_0 = 3$.

EISTI. **Année scolaire 2010-2011**

Département ” Informatique ”

Théorie de l'information

Série d'exercices N°6

Partie I. Codes linéaires correcteurs d'erreurs

Exercice 19. Nous allons expérimenter l'utilisation d'un code de répétition d'ordre 3 pour la transmission d'images en noir et blanc. On suppose qu'une image est une matrice $N \times N$ pixels. Les valeurs des pixels sont 0 (noir) ou 1 (blanc).

Le code de répétition que nous allons étudier consiste à répéter chaque bit trois fois avant de le transmettre. Par exemple, le message 10101 sera transmis sous forme 111000111000111.

On suppose qu'on utilise pour la transmission un canal binaire symétrique de probabilité d'erreur α. Chaque bit a donc la probabilité $1-\alpha$ d'être transmis correctement et la probabilité α d'être inversé.

Le décodage se fait par vote majoritaire. Chaque bloc de 3 bits reçu sera décodé par le bit qui y est majoritaire. La règle de décodage est donc donnée par le tableau suivant :

Bloc reçu	000	001	010	100	101	011	110	111
caractère décodé	0	0	0	0	1	1	1	1

Pour les images, il sera plus pratique de réaliser ce codage de la façon suivante. On transmet la même image trois fois de suite à travers le canal. Ensuite et réalise le vote majoritaire, bit par bit, entre les trois images reçues, pour obtenir l'image d'origine.

Partie I. Expérience. Dans cette partie, nous allons d'abord réaliser quelques expériences à l'aide d'un programme scilab fourni, image.sci. Le programme applique la méthode à une petite image de test, de taille 13×17. L'image est définie par une matrice A dont les valeurs sont binaires. La fonction principale, CTD(A,tauErr), réalise les actions suivantes :

- affichage de l'image d'origine, A
- génération aléatoire de 3 matrices binaires représentant les bits erronés lors de la transmission ; tauxErr est le paramètre qui définit la probabilité d'erreur ;
- calcul de trois matrices transmises en inversant les bits de la matrice originale indiqués par les matrices d'erreur ; cette opération correspond à la **transmission** trois fois de la même matrice à travers le canal ; affichage des trois images reçues ;
- calcul par vote majoritaire bit par bit de la matrice définitive reçue ; cette opération correspond au **décodage** ;
- affichage de la matrice résultat

La fonction renvoie la matrice de cumul d'erreurs. Chaque élément de cette matrice représente le nombre d'erreurs (0-3) pour chaque pixel sur les trois transmissions.

1. Exécuter le fichier image.sci dans scilab. Le taux d'erreur est fixé dans le fichier à 0.1. Est ce que le décodage est parfait ?
2. Essayez la procédure avec un taux d'erreur plus petit (0.05) et plus grand (0.3). Que observez vous ?
3. Comment expliquer que les erreurs subsistent après décodage ? Quel est le lien entre la matrice de cumul d'erreurs par pixel et les pixels qui sont mal décodés ?

Partie II. Analyse 1. Ce code contient deux mots $w_1 = 000$ et $w_2 = 111$. Calculer la distance minimale de ce code et les capacités de détection et de correction ;

2. Est ce que ce code est linéaire ? Si oui, quelle est sa matrice génératrice ?

Exercice 20. Soit un code défini par sa matrice génératrice

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

1. Donner les paramètres (k, n)
2. Ecrire la matrice sous forme systématique
3. Trouver l'ensemble de mots de code (image)
4. Trouver la distance minimale. En déduire la capacité de correction.
5. A partir de la matrice génératrice déterminer la matrice de contrôle
6. On vient de recevoir les mots suivants
   - $m_1 = 00010101$
   - $m_2 = 01000111$
   - $m_3 = 00111010$

   Sans utiliser la table de mots de code comment savoir si ces mots sont des mots de code ?
7. En supposant qu'il y a eu au plus une erreur de transmission par mot, est il possible de retrouver les mots de code transmis ?