TP Théorie de l'Information

March 22, 2022

1 Introduction

Monsieur Pitoune souhaiterait discuter discrètement avec son voisin Boden, mais ne sait pas comment va-t il faire afin de convertir son texte en un format compressible. Il a entendu parler d'un certain codage de source et de canal mais ne comprend rien à ces mots. Il souhaiterait donc passer par un intermédiaire un peu plus qualifié dans le domaine. Il compte sur vous afin que vous puissiez l'aider à s'en sortir. Sur son cahier de charge, il évoque le souhait de pouvoir, avant de faire quoique ce soit, crypter son texte. Ensuite, convertir les caractères cryptés en un format plus adéquat pour la transmission. Et enfin, il souhaiterait que Boden à la réception puisse quand même lire son message malgré d'éventuelles erreurs commises lors de la transmission. A vous de jouer !

PS Créer une classe *Serveur* pour le chiffrage, le codage et la transmission, et une classe *Client* pour la réception, le décodage et le décryptage.

2 Codage de source

La chaine de caractère à transmettre est la suivante :

$$\text{texte} = \text{TESTDETI}$$

1. Comment peut-on envoyer ce texte ?
2. Calculer théoriquement les probabilités pour chacun des caractères. En déduire l'entrope de la source à coder.
3. Coder ensuite une fonction vous permettant d'avoir ces probabilités. Compléter cette fonction afin de pouvoir effectuer le calcul de l'entropie.
4. Vérifier que les résultats obtenus correspondent aux calculs théoriques effectués précédemment.

Afin de pouvoir envoyer ce texte de la manière la plus optimale, on souhaiterait minimiser le nombre d'information envoyé. On propose pour cela d'utiliser le *code d'Huffman*.

5 Dessiner l'arbre de Huffman.

6 Coder l'arbre de Huffman (ce lien va vous aider).

7 En déduire alors le code binaire à transmettre.

8 Coder le calcul de la longueur moyenne des mots de code et le taux de compression.

9 Vérifier l'intégralité des étapes précédentes en testant avec d'autres textes à coder. Comparer les taux de compression. Conclure sur les performances du code de Huffman

3 Codage de canal

Le texte est maintenant sous une forme transmissible. On introduit donc une étape de codage du canal en exploitant les codes correcteurs d'erreur. Pour ce faire, il faudrait donc introduire des informations de contrôle. On donne donc la matrice suivante:

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

10 En déduire à partir de cette matrice le nombre des paquets de mots et des codes à envoyer.

4 Transmission

Afin de simuler une transmission, on souhaiterait implémenter une introduction d'erreur. Pour ce faire, on va choisir, pour chacun des mots, un bit au hasard qu'on va bruiter (le remplacer avec un bit choisit au hasard '1' ou '0'). Dans ce cas là, un erreur peut-être introduite, ou pas.

11 Générer les codes. Concaténer le résultat afin d'avoir le code final à transmettre.

5 Décodage

A l'issue de la simulation précédante, on reçoit un flux de bits. On souhaoterait donc savoir si on a commis une erreur ou pas suite à la transmission.

12 Que proposez-vous ? On donne la matrice H suivante:

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

13 Créer une fonction qui détecte si une erreur a été commise lors de la transmission ou pas.

14 Proposer une correction de ces erreur.

15 Re construiser le message envoyé (chiffré).

16 Décrypter le message.