UPEC
Théorie de l'information
M1DN

Florent Madelaine

2023-24

Master Droit du numérique
Évaluation Formative du lundi 18 mars 2023

Consignes

Durée : 2h30.

Tout document papier autorisé pendant cette évaluation.

Indiquez svp un maximum d'information pour que le correcteur puisse trouver des choses positives dans un travail partiel et/ou partiellement correct.

1 Codes correcteurs

On souhaite envoyer de l'information dans un canal présentant des défauts de transmission. On suppose que ce canal suit le modèle du *canal binaire symétrique* avec une *probabilité d'erreur de un quart* pour chaque bit transmis.

On va envoyer l'information par bloc de deux bits d'information b_1b_2 en codant ce bloc par un mot de code consistant à le répéter 4 fois $b_1b_2b_1b_2b_1b_2b_1b_2$.

Pour rappel, on décode un mot reçu au mot de code le plus proche, c'est-à-dire celui pour lequel il faut changer le moins de bits, quand c'est possible.

1. Si je transmets 0 sur le canal, quel est la probabilité de recevoir 1 ?
2. Si je transmets 1 sur le canal, quel est la probabilité de recevoir 0 ?
3. Sachant que le bit précédent a été mal transmis sur le canal, si je transmets 0 sur le canal, quel est la probabilité de recevoir 1 ?
4. Combien de mots d'information possibles puis je coder ?
5. Combien de mots de code possibles puis je envoyer ?
6. Donnez ces mots de code.
7. Combien de mots puis je réceptionner ?
8. Si je reçois **10101010**, quel est le mot de code le plus proche ? et à quel mot d'information vais-je décoder ?
9. Comment peut-on calculer la distance de Hamming entre deux mots de même longueur ?
10. Quelle est la distance entre le mot **11101010** et les mots du code (pour chaque mot) ?
11. Si je reçois **11101010**, quel est le mot de code le plus proche ? et à quel mot d'information vais-je décoder ?
12. Si je reçois **11111010**, quel est le mot de code le plus proche ? et à quel mot d'information vais-je décoder ?

Correction.

1. Si je transmets 0 sur le canal, la probabilité de recevoir 1 est de 1 quart (probabilité d'erreur d'après l'énoncé).
2. Si je transmets 1 sur le canal, la probabilité de recevoir 0 est de 1 quart.
3. Sachant que le bit précédent a été mal transmis sur le canal, si je transmets 0 sur le canal, la probabilité de recevoir 1 est 1 quart. Dans le modèle du canal binaire symmétrique la probabilité d'erreur est indépendante ldans des bits transmis consécutivement. .
4. Je peux coder $2^2=4$ mots d'information possibles : 00, 01, 10 et 11.
5. Je peux envoyer 4 mots de code possibles (un par mot d'information possible).
6. **00**000000
   01010101
   10101010
   11111111
7. Le mot reçu est de longueur identique à un mot transmis (un mot de code) soit 8 bits. Je peux avoir un nombre arbitraire d'erreurs (de 0 à 8). Je peux donc receptionner n'importe quel mot binaire de longueur 8. Il y en a $2^8=256$.
8. Si je reçois **10101010**, il s'agit d'un mot de code. Le mot d'information correspondant à ce mot de code est 10
9. Pour calculer la distance de Hammiong entre deux mots de mêmùe longueur, il suffit de faire la somme bit à bit modulo 2. Le vecteur obtenu indique 1 à chaque position différente. Le nombre de 1 du résultat (ou poids de Hamming du vecteur) est la distance de Hamming entre les mots puis de regarder
10. Si je reçois **11 10 10 10**, la distance avec : **00**000000 est de 5 (somme modulo 2 donne **11101010**). **01**010101 est de 7 (somme modulo 2 donne **10111111**) **10**101010 est de 1 (somme modulo 2 donne **01000000**) **11**111111 est de 3 (somme modulo 2 donne **00010101**). le mot de code **10**101010
    quel est le mot de code le plus proche ? et à quel mot d'information vais-je décoder ?
11. Si je reçois **11111010**, le mot de code le plus proche est **10101010**. Je décode donc au mot d'information **10**.
12. Si je reçois **11111010**, le mot de code le plus proche n'est pas unique **11**111111 et **10**101010 sont à distance 2. Les autre sont à distance 6. Il n'y a donc pas de mot d'information vers lequel il soit raisonnable de décoder.

2 Codes correcteurs (suite)

On considère le protocole de codage qui consiste à coder deux bits b_1b_2 en $b_1b_1b_1b_1b_2b_2b_2b_2$. On parle du *protocole non entrelacé* dans ce cas et de *protocole entrelacé* pour celui de l'exercice

précédent.

1. Combien de mots d'information possibles puis je coder avec le protocole non entrelacé ?
2. Combien de mots de code possibles puis je envoyer avec le protocole non entrelacé ?
3. Donnez ces mots de code.
4. Que pensez vous du *protocole entrelacé*, par rapport au *protocole non entrelacé* (argumentez) ?
5. On change le modèle du canal qui reste symmétrique (l'erreur ne dépend pas du fait qu'on lis un 1 ou un 0) mais dorénavant, les erreurs simulent un problème local physique (rayure sur un support cd ou poussière ou pic d'interférence) et on perd l'indépendance. On va tirer au hasard avec une probabilité d'un huitième d'avoir une erreur sur un bit. Par contre, si il y a une erreur sur un bit, on va toujours ajouter une erreur soit à gauche, soit à droite de ce bit.

   Avec ce nouveau modèle d'erreur, que penseez vous du protocole entrelacé vs celui qui n'est pas entrelacé (argumentez) ?

Correction.

1. Je peux coder $2^2 = 4$ mots d'information possibles : 00, 01, 10 et 11.
2. Je peux envoyer 4 mots de code possibles (un par mot d'information possible).
3. **0**0000**0**000
   00001111
   11110000
   11111111
4. Le protocole non entrelacé n'apporte rien de plus dans le cadre du modèle d'erreur du canabl binaire symmétrique. Si on réordonne les bits la distribution des erreurs est identique.

   Si on note k_1 le nombre d'erreurs de transmission sur les 4 premiers bits, et k_2 le nombre d'erreurs de transmission sur les 4 derniers bits, alors l'analyse de ce qui se passe peut-être faite indépendamment sur chaque bloc de 4 bits, comme si on transmettait un bloc de 1 bit répété 4 fois. Si k_i vaut 0 il n'y a pas d'erreur et c'est correct. Si k_i vaut 1, l'erreur est detectée et la correction est correcte. Si k_i vaut 2, l'erreur est detectée et la correction n'est pas possible. Si k_i vaut 3, l'erreur est detectée et la correction est incorrecte. Si k_i vaut 4, l'erreur n'est pas detectée.

   L'entrelacement ne change rien à cette analyse. On peut définir similairement k_1 et k_2 comme le nombre d'erreur sur les 4 bits correspondant à b_1 (les bits impairs) et b_2 (les bits pairs). L'analyse reste identique et ne dépend que de la valeur de k_i.
5. Si On change le modèle du canal comme indiqué dans l'énoncé, alors le code entrelacé devient pertinent. En effet, le nombre d'erreur n'est pas vraiment plus grand en moyenne mais sont proches.

   Dans le cas non entrelacé on va détecter 2 erreurs et on ne pourra pas décoder. Dans le cas entrelacé, les 2 erreurs vont concerner 2 blocs différents, et on peut corriger 1 erreur correctement.

3 Compression

On souhaite compresser le texte ci-dessous avec la méthode de Huffman.

```
babar_va_au_bar.
```

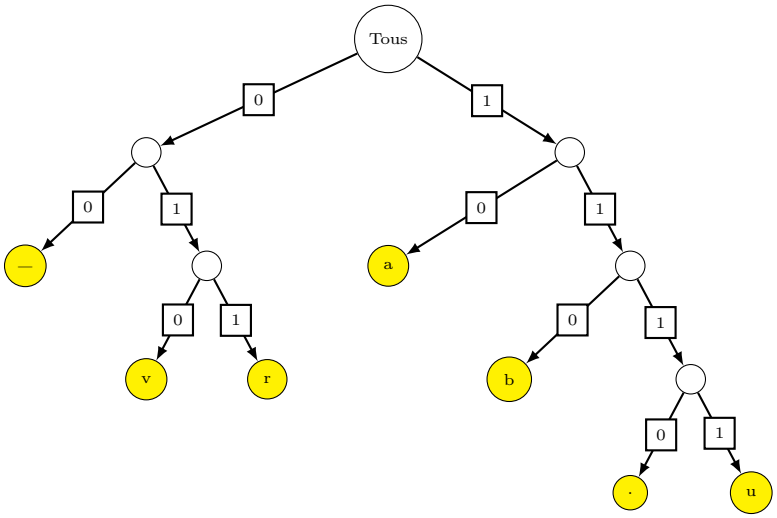
NB. pour faciliter le travail de correction, la file d'attente départagera les arbres de même poids de la manière suivante : en priorité les arbres les plus courts et en cas d'arbres de même poids et de même hauteur, ceux qui ont le moins de feuilles. Si il y avait encore des cas à départager, on considérera comme prioritaire l'arbre contenant la plus petite feuille pour l'ordre "ascii-bétique". Dans cet ordre, l'espace (noté _ ci-dessus) vient en premier puis le point, puis les lettres dans l'ordre alphabétique.

1. Quelle est la file d'attente au début de l'algorihme ?
2. Calculez toutes les étapes de construction de l'arbre de Huffman en indiquant bien à chaque fois l'ordre de la file d'attente.
3. Quel dictionnaire faisant correspondre symbole et mot binaire obtient-on à partir de l'arbre de Huffman ?
4. Expliquez comment les 6 premières lettres du texte sont codés en binaire par la méthode de huffmann pour cet arbre.

4 Décompression

On souhaite décompresser un texte binaire obtenu par une méthode de Huffman avec l'arbre suivant.

l'arbre :



1. Donnez le dictionnaire pour cet arbre.
2. Qu'est-ce qui permet de décoder sans problème le texte binaire alors que les mots sont de longueurs différentes ?
3. Décodez le texte binaire ci-dessous en expliquant bien votre démarche.

   `10111100110100110011010110100110001010`

4. Si la transmission s'interrompait avant la fin du texte, quel problème potentiel aurait-on ?

Correction.

1. Le dictionnaire est obtenu en lisant le chemin depuis une feuille vers la racine.
   Dans l'ordre des feuilles :
   - _ : 00
   - v : 010
   - r : 011
   - a : 10
   - b : 110
   - . : 1110
   - u : 1111
2. Ce qui permet de décoder sans problème le texte binaire alors que les mots sont de longueurs différentes (pas d'alignement) et que nous n'utilisons pas de séquence particulière pour servir de séparateur c'est que nous avons un *code préfixe*. C'est-à-dire qu'aucun mot du code n'a pour préfixe un autre mot du code.
3. Décodez le texte binaire ci-dessous en expliquant bien votre démarche.
   On découpe le texte de la gauche vers la droite en identifiant les mots de code et en les remplaçant par la lettre correspondante à l'aide du dictionnaire. En pratique pour décoder, l'arbre suffit.

10	1111	**00**	110	**10**	011	**00**	110	**10**	110	**10**	011	**00**	010	**10**
a	u	_	b	**a**	r	_	b	**a**	b	**a**	r	_	v	**a**

   Le message est donc :

   `au_bar_babar_va`
4. Si la transmission s'interrompait avant la fin du texte, on pourrait décoder sans soucis le début du message. On aurait un souci potentiel à la fin (mot de code partiel) ce qui permettrait de détecter le problème.