Modélisations mathématiques R5B10
Dossier : Théorie de l'information, algorithmes de compression.

Dans un texte, toutes les lettres ne sont pas forcément importantes pour la compréhension. On peut définir dans un texte (ou dans un autre événement) l'entropie qui lui est associée, qui mesure l'information apportée par chaque lettre (ou chaque mot, ou chaque élément...). La théorie de l'information permet de formaliser cette notion, avec de nombreux domaines d'application.
Nous proposons dans ce dossier d'aborder certaines notions de ce domaine.

Entropie, ou mesure de l'information
Les travaux de Shannon, en 1948, ont posé les fondements de ce qui est la théorie de l'information. La question est de mesurer la quantité d'information apportée par la donnée de la réalisation d'un processus (considéré comme un processus aléatoire).
Par exemple, si on lance une pièce de monnaie (non truquée) et qu'on regarde sur quelle face elle tombe, l'information apportée est moindre que si on lance un dé (non truqué) et qu'on regarde le résultat. En effet, dans le premier cas on connaît la réalisation qui est advenue parmi 2 possibles, dans le 2e cas c'est parmi 6 possibles.
D'une manière générale, si on considère qu'une source est un processus aléatoire comportant n symboles, le symbole i ayant une probabilité p_i d'apparaître, alors l'entropie de la source est

$$H = -\sum_{i=1}^{n} p_i \log p_i.$$

Si log est le $\log_2$ alors l'unité est le *bit* par symbole. Pour mesurer l'entropie d'une réalisation donnée, on remplace dans la formule ci-dessus la probabilité p_i par la fréquence observée f_i.

Code de Huffmann
Pour compresser un texte (ou un autre objet) sans perte, on peut espérer au mieux le décrire avec autant d'information que son entropie. Un algorithme qui réalise la compression sans perte est le *codage entropique* (ou codage de Huffmann, du nom de son inventeur).
L'idée est de représenter chaque symbole du texte (ou chaque mot) par un code, les codes les plus courts servant à représenter les symboles les plus fréquents. Pour qu'il n'y ait pas d'ambiguïté au décodage, il faut qu'aucun mot du code ne soit préfixe d'un autre mot du code.
Cette implémentation se fait à l'aide d'un arbre binaire. On pourra se reporter à la page wikipédia citée pour les détails (celle en anglais est plus complète).

Travail demandé

- Vous approfondirez les notions présentées ici à l'aide des références ci-dessous et de ce que vous pourrez trouver par vous-même.
- Calculer l'entropie des différents textes trouvés dans les fichiers `texte1.txt, texte2.txt`, en considérant chaque **lettre** (y compris les espaces) comme un symbole. Même question pour `texte2.txt` en considérant chaque **mot** comme un symbole.
- Calculez l'entropie de différents textes (suffisament longs, trouvés à votre initiative) que vous calculerez 1) avec la fréquence d'apparition des lettres 2) avec la fréquence d'apparition des mots.
- Vous implémenterez un codage de Huffmann dans le cas d'un texte formée de n lettres, la lettre i ayant une fréquence d'apparition p_i.
- Appliquez ce code dans un texte aléatoire de grande longueur $\ell = 10000$ (par exemple) avec $n = 3$ (3 lettres A, B, C). Testez le résultat avec différentes fréquences pour les lettres.
- Appliquez ce code à des textes (suffisamment longs) trouvés à votre initiative.
- Vous rédigerez un rapport présentant votre travail.

Références

- `https://fr.wikipedia.org/wiki/Théorie_de_l'information`
- `https://fr.wikipedia.org/wiki/Entropie_de_Shannon`
- `https://en.wikipedia.org/wiki/Huffman_coding`
- générateur de texte aléatoire : `https://fr.lipsum.com/`