Notes de cours d'informatique théorique (HMIN 118)

Guillaume Pérution-Kihli

31 octobre 2018

Sommaire

1 Théorie de l'information de Shannon

1.1 Mesure de l'information de Hartley

L'information est le nombre de réponses possible.

Hartley suppose que pour composer un message, il faut avoir un alphabet dans lequel le composer. Cet alphabet doit comporter un nombre fini s de symboles. Un message comportant n symboles tirés d'un alphabet de taille s n'est donc qu'un parmi les s^n messages possibles.

D'après Hartley, la quantité d'information intrinsèque au message doit être proportionnelle à sa longueur fois une quantité qui ne dépend que du nombre de symboles de l'alphabet. Autrement dit, la quantité d'information H d'un message de longueur n est de la forme $H = k * n$ où la constante k est à déterminer.

Pour obtenir la fonction désirée, Hartley propose d'utiliser le logarithme et de calculer

$$H = log(s^n) = n * log(s)$$

.

Ce logarithme peut avoir différentes bases, et l'unité de mesure de l'information change en fonction de ces bases.

base		nom de l'unité
2	$2(log_2)$	bit
e	$e(ln)$	nat
10	$10(log_{10})$	hartley

Cette méthode de mesure a cependant un important défaut : tous les symboles du message sont considérés comme contenant la même quantité d'information puisqu'on les multiplie tous par la même constante.

1.2 Mesure de l'information de Shannon

Pour Shannon, l'unité d'information est le bit. Avec le bit comme mesure, l'information devient une notion essentiellement probabiliste donc quantifiable. Et s'il y a beaucoup de manières de passer un message en binaire, il n'y en a qu'une qui permette de réaliser la quantité d'information de Shannon. L'entropie de Shannon mesure la quantité d'information d'un message : un signal peu informatif est redondant et donc prédictible d'un point de vue des probabilités, alors qu'un signal très informatif est très diversifié et donc peu prédictible.

Pour une source, qui est une variable aléatoire discrète X comportant n symboles, chaque symbole x_i ayant une probabilité P_i d'apparaître, l'entropie H de la source X est définie comme :

$$H_b(X) = -\mathbb{E}[\log_b P(X)] = \sum_{i=1}^{n} P_i \log_b \left(\frac{1}{P_i}\right) = -\sum_{i=1}^{n} P_i \log_b P_i.$$

où $\mathbb{E}$ désigne l'espérance mathématique, et $\log_b$ le logarithme en base b. On utilise en général un logarithme à base 2 car l'entropie possède alors les unités de bit/symbole. Les symboles représentent les réalisations possibles de la

variable aléatoire X. Dans ce cas, on peut interpréter $H(X)$ comme le nombre de questions à réponse oui/non que doit poser en moyenne le récepteur à la source, ou la quantité d'information en bits que la source doit fournir au récepteur pour que ce dernier puisse déterminer sans ambiguïté la valeur de X.

$$H(X) = H_2(X) = -\sum_{i=1}^{n} P_i \log_2 P_i.$$

Exemple 1.2.1. Prenons $X = yabadabadoo$. On obtient la répartition qui suit :

Symbole	a	b	d	o	y
Nombre d'occurrences	4	2	2	2	1
Probabilité d'apparition P_i	$\frac{4}{11}$	$\frac{2}{11}$	$\frac{2}{11}$	$\frac{2}{11}$	$\frac{1}{11}$

L'entropie de X est :

$$H(X) = -\sum_{i=1}^{5} P_i \log_2 P_i = -(\frac{4}{11}\log_2\frac{4}{11} + 3 * (\frac{2}{11}\log_2\frac{2}{11}) + \frac{1}{11}\log_2\frac{1}{11}) \simeq 2.19$$

2 Codes et techniques de compression

2.1 Codes, codes préfixes et inégalité de Kraft

Scénario 2.1.1. A veut transmettre quelques informations à B.

$element \in \chi \rightarrow chaine\ binaire$

Définition 2.1.1. Fonction de décodage

$D : 0, 1^* \rightarrow \chi$.

$E = D^{-1}$.

Théorème 2.1.1. Inégalité de Kraft

Il existe un code uniquement décodable sur un alphabet de taille r, avec n mots de code de tailles $l_1, l_2, ..., l_n$, ssi

$$\sum_{i=1}^{n} r^{-l_i} \leqslant 1$$

Définition 2.1.2. Sans préfixe

Un code est dit sans préfixe si aucun mot de code n'est préfixe d'un autre.

2.2 Codage de Shannon-Fano

Le codage de Shannon-Fano est un algorithme de compression de données sans perte élaboré par Robert Fano à partir d'une idée de Claude Shannon. Il s'agit d'un codage entropique produisant un code préfixe très similaire

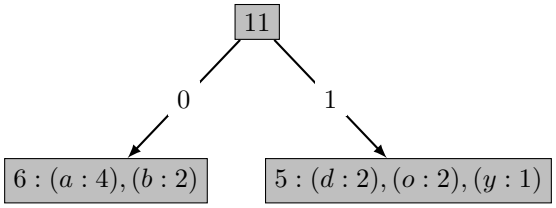
à un code de Huffman, bien que pas toujours optimal, contrairement à ce dernier.

Algorithme 1 : Algorithme de Shannon-Fano

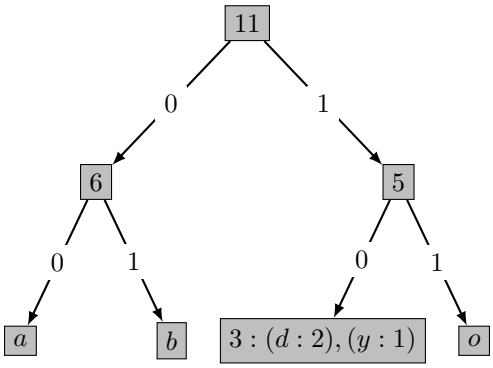
1 Pour chaque symbole, compter la fréquence d'apparition;
2 Les ordonner des plus fréquents aux moins fréquents ;
3 Diviser la liste de symboles en deux parties de façon à avoir des totaux proches ;
4 Assigner 0 à la partie gauche de la division, et 1 à la partie droite ;
5 Recommencer les étapes 3 et 4;

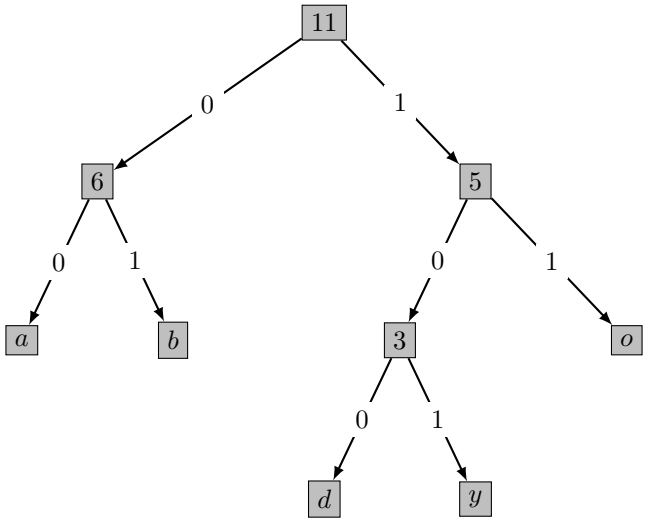
Exemple 2.2.1. Reprenons l'exemple 1.2.1. On obtient la liste de symboles ordonnés : $< (a : 4), (b : 2), (d : 2), (o : 2), (y : 1) >$.

On va diviser cette liste une première fois en deux listes ayant à peu près les mêmes nombres d'occurrences, qu'on va représenter sous forme d'arbre :



On répète le processus jusqu'à obtenir un arbre où chaque feuille représente un symbole :





On obtient ainsi des codes sans préfixe pour chaque symbole.

Symbole	a	b	d	o	y
Code	00	01	100	11	101
Taille code	2	2	3	2	3

La longueur totale de l'encodage de la chaîne sera : $2*4+2*2+3*2+2*2+2*1=25$ bits. Le nombre de symboles moyen encodés par bit est de $\frac{25}{11} \simeq 2,27 > 2,19 \simeq H(X)$.

2.3 Codage de Huffman

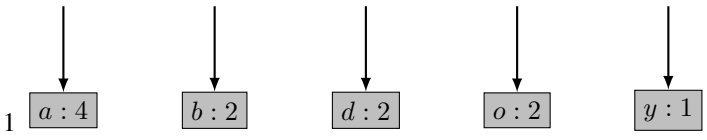
L'approche du codage de Shannon-Fano est descendante : l'algorithme part de l'ensemble des symboles et divise cet ensemble récursivement jusqu'à arriver à des parties ne contenant qu'un seul symbole. L'inconvénient de cette approche est que, lorsqu'il n'est pas possible de séparer un ensemble de symboles et deux sous-ensembles de probabilités à peu près égales (c'est-à-dire lorsque l'un des sous-ensembles est beaucoup plus probable que l'autre), les codes produits ne sont pas optimaux.

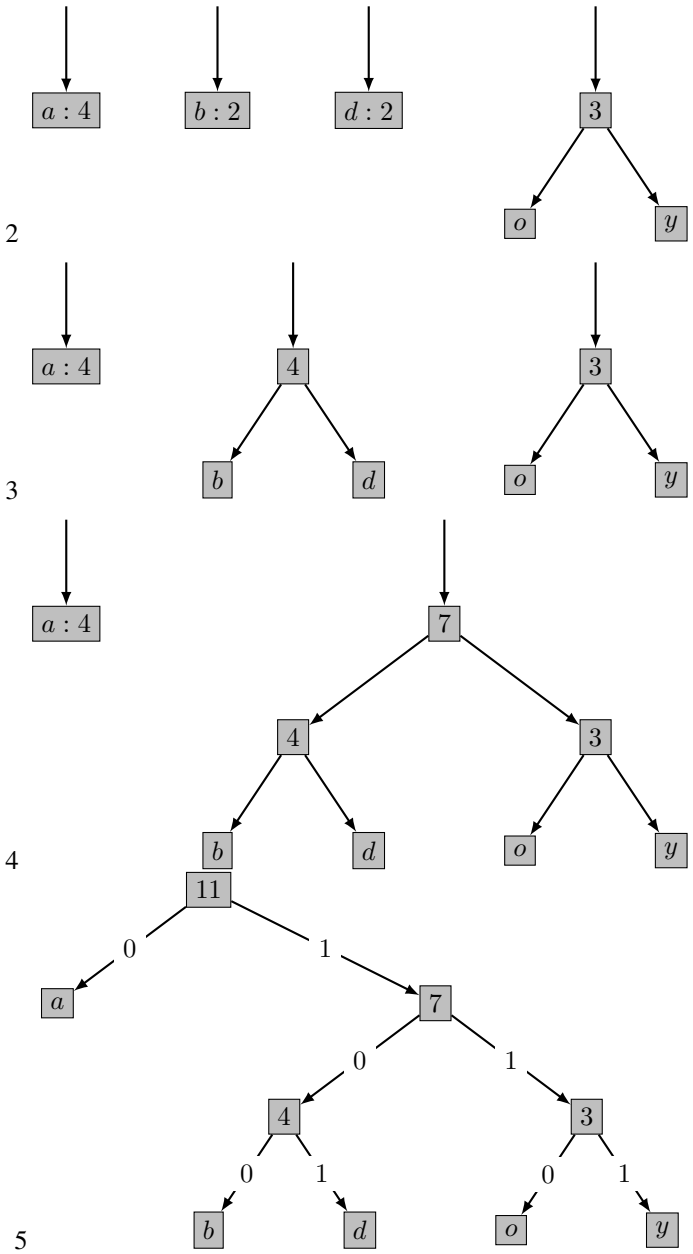
Le codage de Huffman a une approche ascendante : l'algorithme part des symboles et regroupe ceux ayant la probabilité la plus faible, jusqu'à avoir regroupé tous les symboles. Cette approche permet d'obtenir systématiquement un code optimal au niveau du symbole, dans le pire cas de la même longueur que le code de Shannon-Fano équivalent, dans tous les autres cas plus court.

Algorithme 2 : Algorithme de Huffman

1 Pour chaque symbole, créer une feuille, l'associer avec sa fréquence d'apparition et l'ajouter à une file de priorité;
2 **tant que** *il y a plus d'un nœud associé à une fréquence dans la file* **faire**
3 Supprimer de la file les 2 nœuds ayant la fréquence la plus faible;
4 Créer un nouveau nœud ayant pour fils les 2 nœuds supprimés, l'associer à la somme des fréquences des nœuds fils;
5 Ajouter le nouveau nœud à la file;
6 **fin**
7 Le nœud restant dans la file est la racine de l'arbre;

Exemple 2.3.1. Reprenons l'exemple 1.2.1 avec $X = yabadabadoo$. Les étapes de construction de l'arbre de Huffman sont les suivantes :





On obtient ainsi des codes sans préfixe pour chaque symbole.

Symbole	a	b	d	o	y
Code	0	100	101	110	111
Taille code	1	3	3	3	3

La longueur totale de l'encodage de la chaîne sera : $1 * 4 + 3 * 7 = 25$ bits. Le nombre de symboles moyen encodés par bit est de $\frac{25}{11} \simeq 2,27 > 2,19 \simeq H(X)$. On constate qu'on est sur un cas avec la même moyenne que Shannon-Fano.

Annexe

Sources

- Mesurer l'information selon Hartley
- Entropie de Shannon
- Claude Shannon : Le monde en binaire
- Kraft–McMillan inequality
- Codage de Shannon-Fano
- Shannon–Fano coding